Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

# MASTER THESIS

**TITLE: Industry 4.0: Application of advanced services in logistics**

**MASTER DEGREE: Master's degree in Applied Telecommunications and Engineering Management (MASTEAM)**

**AUTHOR: Edgar Gustavo Delgado Tello.**

**ADVISOR: Alcober Segura, Jesús**

**DATE: September, 6th 2018**

**Title:** Industry 4.0: Application of advanced services in Logistics

**Author:** Edgar Gustavo Delgado Tello

**Advisor:** Alcober Segura, Jesús

**Date:** September, 6th 2018

## Abstract

Industry 4.0 is a concept that refers to the fourth industrial revolution and involves an intelligent interconnected industry making use of digital technologies (IoT, Big Data, ERP, Cloud Computing, Business Intelligence, etc.) allowing data extraction in real time and its control remotely.
Logistics is an area that stands out in this environment of Industry 4.0, which is a key element to optimize the supply chain of factories and small companies.

The goal of this master thesis is to make an overview of the state of Industry 4.0, its default protocol of communication OPCUA and to show that is possible to democratize the use of Industry 4.0 technologies making a proof of concept of Vertical Communication between a Logistic Tracking Elements of a Logistic Chain (NFC Tags) with the management side (ODOO software) as a solution for inventory control for small and medium companies. The potential of the business model is briefly reviewed.

# CONTENTS

# INTRODUCTION

The Industry 4.0 start as a German Initiative, thinking in the future and how the European Manufacturers can maintain its market, boosting the Industry and Competitiveness without losing space Abroad. This concept of Industry 4.0 refers to the 4rth Industrial Revolution and integrate many technologies among which are: Big Data, ERP, Cloud Computing, Business Intelligence, etc, with the main goal of first have a global Interconnection between all the layers of the Enterprise, from the Control Systems to the Upper Layers (Decision Takers Layers) and to extract and process all the information to improve the performance of the whole system.

OPC UA has been chosen as the standard protocol to be used and provide Interoperability in the Industry 4.0 for vertical and horizontal communication for cyber physical systems. This protocol has been chosen as a standard for its safe, scalability, reliability and interoperability between different industrial protocols and manufacturers, allowing an easy integration with an open architecture.

The Logistics plays an important role in the production lines and in the actual business scenarios because could define the effectiveness of a product or service, the final price and could influence in the decisions taken, so is important to have and integration and communication between all the levels in the Company.

A problem has been identified in small and medium-sized companies, the inventory control. This task is very critical and it is done manually, which represents a lot of time invested and the possibility of making mistakes when doing it. It is proposed the design and testing of a proof of concept of vertical integration between an ERP business management software ODOO with an NFC reader for inventory control with TAGS using the OPCUA protocol for the communication.

This Master Thesis is divided in the following Chapters:

- Chapter one gives an introduction reviewing the objectives and motivation for doing this work.
- Chapter two highlights the Industry 4.0, the concepts, the state of the art behind it and the importance and technologies used by Logistics.
- Chapter three focus on the design, implementation and test of the proof of concept.
- Chapter four the conclusions.

# CHAPTER 1. OBJETIVES

## 1.1. Objectives

The Objective of this Master Thesis is to review the state of the art of the Industry 4.0 with their technologies behind that are the key point for its working, its standard protocol of communication OPCUA as well as the development of a Proof of Concept of Vertical Integration between the ERP Software ODOO and a NFC Reader for Inventory Control using the OPCUA protocol, finally the feasibility of implementation of the Industry 4.0 will be discussed based on the experience obtained after carrying out the project.

## 1.2. Motivation

The massive use of Internet and the integration of technologies as the IOT, Big Data, Data Analytics, ERP, etc allows to have a great amount of resources and information which working all together in Technologies such as Industry 4.0 offer an advantage for optimization and decision taking in the Industry.

Actually, there are big companies such as DHL that invest a lot of money to develop solutions Ad-hoc that fits their needs, but the main goal nowadays is to democratize the use of the technologies and solutions to small companies and start-ups in a way that they can integrate their ERP and management systems with the industrial and operative plant, sharing information between platforms, what will allow them to have the control, take fast decisions and adapt easily to the market needs. At this point appears OPCUA as the protocol chosen that will allow this integration. OPCUA is mostly used for horizontal Integration between devices and applications from different manufacturers, but exists SDK's and library's available for different development platforms that will allows to implement vertical integration with the upper management layers, therefore in this work a proof of concept between ODOO and an NFC reader for Inventory Control will be performed using OPCUA.

# CHAPTER 2. INDUSTRY 4.0

## 2.1. State of the Art

This section will explain the evolution of industrial revolutions from Industry 1.0 to Industry 4.0, then details the advantages and benefits of the implementation of the Industry 4.0 as well as the main technologies that are part of it, then the protocol chosen by default OPC UA is explained with its importance, architecture, technology and a brief explanation of the requirements and procedures to integrate and OPC UA device into an application, the NFC protocol is explained with its different modes of communication and operation, its advantages and uses and finally a brief review of the ODOO ERP software is performed.

### 2.1.1. Industrial Revolutions

Throughout the history, there have been great advances and discoveries that changed the way in which the industry performed. These changes improved the productivity and the way of living of the people, existing three previous revolutions that involve the use of different technologies.

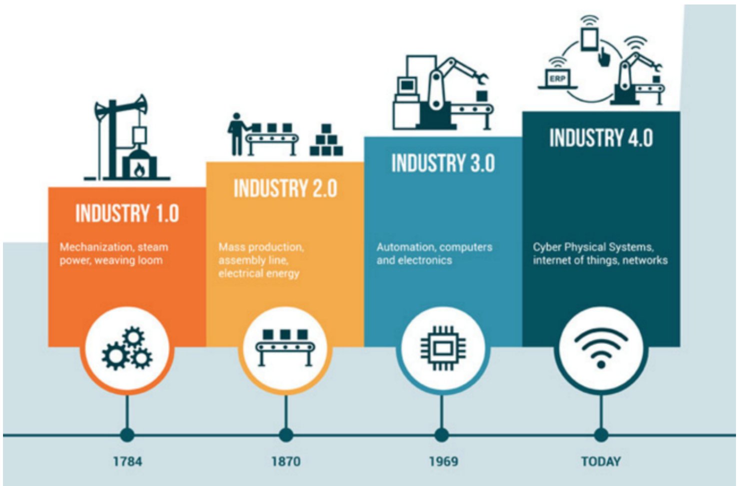



**Fig. 2.1** Evolution of the Industry

The First Revolution (1784 – middle of XIX Century) did the Mechanization of Manufacturing with the Introduction of Steam and Water Power, the second revolution (late XIX Century-1970) start the mass production assembly lines using the electrical power. The third revolution (1970-nowadays) uses an automated production, with the implementation of Electronics, PLC, Robotics, IT Systems, and the Fourth Revolution is been developed actually, also known as the smart factory, implements technologies as Machine Learning, Big Data, IoT to personalize and produce in a more efficient way.

### 2.1.2. Industry 4.0

The Industry has a very important role in the community, is a fundamental pillar for the labour and the economy market, being a concern for the governments to keep it improving. In this Context the government of Germany as a strategy for the German Industry start using the term “Industry 4.0” from 2010 and in Spain was announced in 2014 [1]. This concept introduce the use of some technologies as IoT and Internet Services in the Industry allowing to co-work the physical world (devices, materials, machines) with the digital systems. The motivation for introducing this concept is:

- Increase the Industrial Added Value and employment in the sector [1].
- Improve the Competitive of Industrial production due to low cost production of emerging countries.
- Develop an Industry that can easily adapt to the customer’s needs, customising the products and manufacturing as fast as possible.

The Industry 4.0 will make increase the production, and act as a bridge that joins the Information Technology and the Industry itself. The essence is the ubiquitous connection of machines, products and people allowing a communication between them. A negotiation between tools, products, means of transport is expected within a virtual marketplace creating a link between the virtual and the physical world [2].

The Main Idea behind this concept is that the machines can decide when is better to consume the energy making use of many technologies such as Big Data, IoT, Augmented Reality, Beacons, RFID, NFC, etc.

The Industry 4.0 (Smart Factory) requires a vertical integration from the Physical Entities to Information systems [3]. That means that all the devices in the Factory Floor (PLC, Machines, Scada’s, Robots) should be able to communicate and take decisions with the upper layers of the organization that includes the plant level (MES) and the enterprise level (ERP).

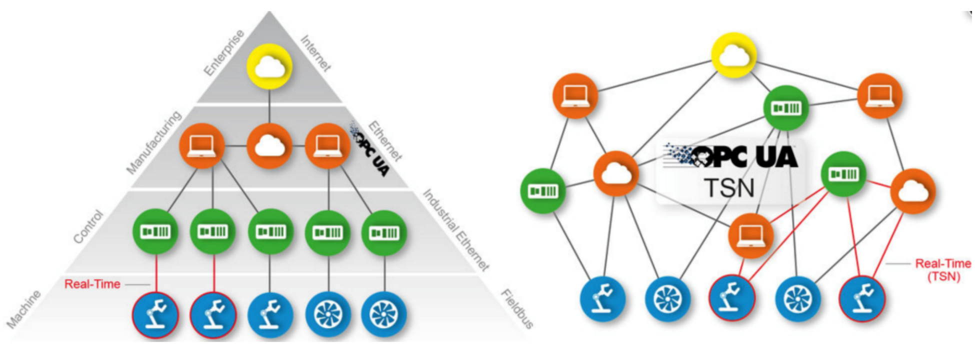



**Fig. 2.2** Automation Pyramid [4]

Based on this necessity of communication with a standard protocol, to make possible the integration of all devices from different manufactures, OPC UA was chosen as the protocol that allows the Industry 4.0 to be developed and implemented in the Smart Factories [5].

The reasons it was chosen as the default protocol, are because meet the challenges posed by Industry 4.0 as is security, standardized exchange of data and information between devices, machines and services across different industries. In November 2016 was published a classification for the products as Basic, Ready and Full, even the lowest requirements to be addressable via TCP/UDP and to integrate at least the OPC UA information model [5].

The Implementation and use of Industry 4.0 will bring many benefits that can be grouped in the following categories according to [6]:

- Productivity: Millions of Euros of Economic Benefits in the following years, especially in the Parts Manufacture and Automotive Industry.
- Revenue Growth: Germany predicts and increase of 1% in PIB due to an increase of demand of Improved Equipment's and new data applications.
- Employment: An increase of Employment is expected although the need for different skills. Low-Skilled workers can be displaced by machines and Mechanical Engineers, Software Developers and IT Experts will be in a great demand.
- Inversion: Investment will be needed to adapt the facilities and Technology implementation.

## 2.2. Main Components of Industry 4.0

The Industry is digitizing in an exponential way, in parallel with technologies that are showing a constant evolution as they are: sensors, robots, internet of things, big data, augmented reality, NFC, which produce an increase in automation for the industries.

Industry 4.0 integrates the technologies of the third industrial revolution together with components for information processing that allows to store, process and manage the information. These components are the following: Big Data, Internet of things, Internet of services and Cyber-Physical Systems [7]. In this sections the components will be described briefly.

### 2.2.1. Big Data

Big Data is a term that refers to management and analysis of big amount of information (Structured and Unstructured) that cannot be analyzed using regular methods. The information comes from different sources and is used to gain insight and to make strategic decisions that allows to gain value in the business.

The increasing number of connected devices increase the amount of information available from different sources ( sensors in machines, wearable devices, social media, e-commerce, etc.) make necessary the analysis of this information to help to gain more value to the businesses and to improve many aspects like the sales, preventive maintenances, customer service [8].


**Fig. 2.1** Big Data [9]

### 2.2.2. Internet of Things

Internet of things can be defined as a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies [8]. It is also defined as the connection to the internet of a massive amount of inexpensive machines coupled to daily life things [10].

The number of interconnected devices is increasing and there is a great potential to generate a great economic impact in the industries. In the following graph is possible to see some growth projections of connected devices according to the MIT technologies magazine.

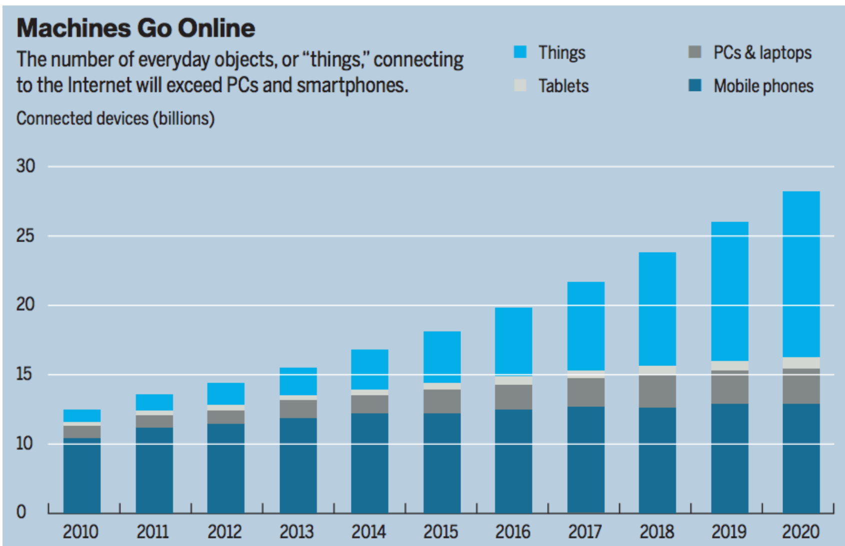



**Fig. 2.2** IOT Projections [11]

### 2.2.3. Internet of Services

Apart from having an interconnection, the manufactures and Industries need to think through their business models. The internet of services refers to the way in that vendors offers their services, it consist or is made of the infrastructure for the services, the participants, the business models and the services itself [2].

### 2.2.4. Cyber-Physical Systems (CPS)

The Cyber Physical Systems refers to the integration of computation, networking and physical processes, meaning that the production is accompanied by computed based processes with ubiquitous computing. Includes sensors and actuators to collect and send data [2].

## 2.3. OPC UA

It will be explained the importance of the OPC UA protocol, then continue with a view of Legacy OPC and its improvements in OPC UA, continuing with the characteristics, architecture, specifications, technology, procedures and requirements to integrate an OPC UA device into an application.

### 2.3.1. Importance of the protocol.

One of the great concerns in the implementation of the idea of Industry 4.0 is the exchange of information and data in a secure and standardized way between the different devices and layers of the components in the Industry, so in April 2015 was recommended as the Reference Architecture Model for the Industry 4.0 the IEC Standard OPC UA (Unified Architecture). The devices integrated to this Industry 4.0 Environment, even the basic ones should be addressable via TCP/UDP and Integrate the OPC UA information model (Integrated or via gateway) [5].

OPC UA protocol will be explained starting by a brief view of the Legacy OPC Standard to continue with the new improvements of the OPC UA protocol.

### 2.3.2. Legacy OPC

OPC are a set of standards developed by the OPC Foundation for Control and Monitor of Industry Equipment, this standard was released in 1996 as a solution resolving interoperability problems between the different devices. The standard is based in the Microsoft DCOM Technology developed in 1990 to operate in Windows Operative Platforms. The base of this protocol is the Client-Server Architecture. The protocol itself defines the way that OPC-Client / OPC-Server communicate and exchange information

There are some great Limitations of OPC that press the development of the new implementations:

- Platforms Limitation: Originally was developed to work with Windows Platforms leaving aside other systems as UNIX, LINUX, MAC OS [12].
- Remote Data Access Limitations: Data cannot be transferred between different Networks making that higher layers cannot access to production data for low layers to making decisions [12].
- Low Security and a Complex Configuration to manage the information, is complex to connect through a Firewall.

OPC standard was the base for OPC Foundation to develop four types of OPC servers and are the following:

- Server OPC DA (Data Access). Designed to transmit information in real time.
- Server OPC A&E (Alarms and events). Send Alarms and events to the client.
- Server OPC HDA (Access to Historical Data). Bases in the access of the client to historical Data.
- Server OPC UA (Unified Architecture). This Specification allows working with any type of Data.

### 2.3.3. OPC UA

OPC UA the new version of the OPC Servers, it was released in 2008 and makes some improvements in the limitations of OPC. This version stop using DCOM interface for its communication and this allow not only to work with windows systems but also to many other operative systems, that opens the possibility of interaction to many new devices.

OPC UA was born out the desire to create a true replacement of COM-based specifications without losing features or performance [13].

Some features of OPC UA are :

- This protocol implemented an address space and a service model integrated, that permit to access to data alarms, historic data form its own address space using the same set of services. The data is presented hierarchically.
- OPC UA offer support to SDKs in JAVA, C++, C# and other programming languages, in this way there are no limitations of the operative system.
- Integrates the characteristics of its predecessors, historic information and actual data are saved in variables and the control commands for the devices connected are managed by methods in the same server.
- For flow of information OPC UA improves the security system, now implements the use of certificates and public keys using a secure messaging environment through TCP and HTTP.
- The Authentication process for each client and server is made using OpenSSL.

- The protocol presents many different features to be used by servers and each can implement the ones that are necessary for its use.
- The protocol added data models and semantic models to deal with complex data structures to form new data models and deal with complex data structures and the clients don't need to understand the different data to identify [12].
- OPC UA is independent of the vendor or system that produces the application, and the communication is independent of the programming language [14].

### 2.3.4. Architecture of OPC UA

OPC UA protocol is based in client-server model so in this way the device that needs to post information is called UA-Server and the devices that need to use the information is called UA-Client.

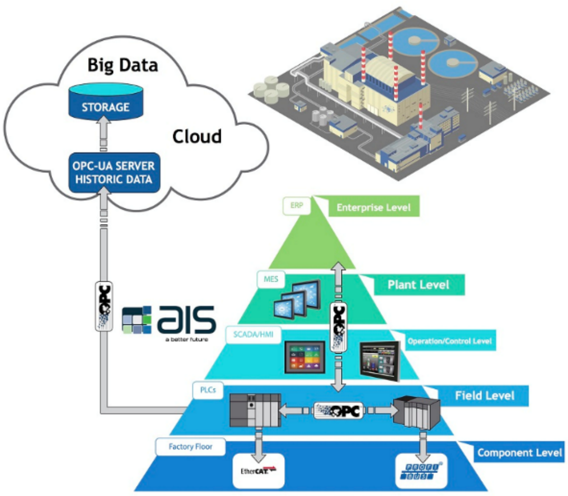



**Fig. 2.3** OPC UA in the Organization [15]

Each level of the organization can contain multiple Clients and Servers, each client can be connected to many servers at the same time and work in different Operative Systems (Windows, Linux) having a complete interoperability between all. The application can be working in the server and client side.

#### *2.3.4.1. OPC UA Client*

The OPC UA Client includes OPC UA Client Structure, OPC UA Client Application, OPC UA Client Communication Stack and OPC UA Client API to be in contact with server [12]. The communication Stack convert the API calls in messages. The Stack obtains responses and notifications messages from the server and passes messages to the client application through API.

### 2.3.4.2. *OPC UA Server*

The OPC UA Server includes OPC UA Server Application, real objects, OPC UA Address Space, OPC UA Server Stage API and the Communication Stack [12]. The Address Space is composed of nodes and is used to represent real objects and the clients through OPC Interfaces can access to the nodes and services.

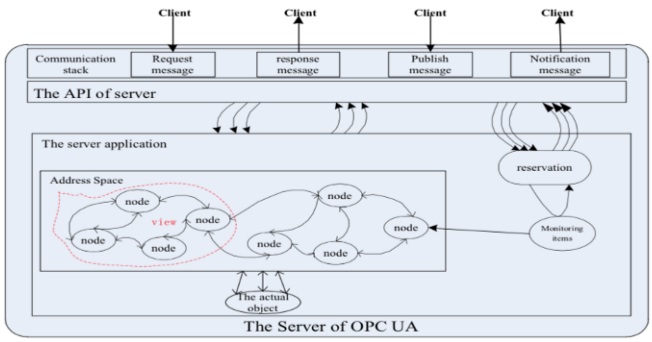



**Fig. 2.4** OPC UA Server [12]

## 2.3.5. OPC UA Specification

OPC UA has a series of specifications, known also as IEC 62541 Standards. They are split into core specifications and define protocol for OPC UA and how is the access in the information models.

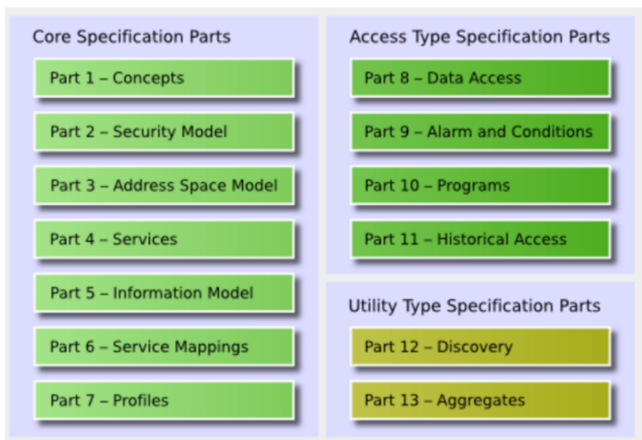



**Fig. 2.5** OPC UA Specification. [16]

In the figure 2.5 we can appreciate the Specification model. The Part 1 (Concepts) gives an overview of OPC, Part 2 describes the security model. The Part 1 and Part 2 can be modified depending the requirements of the application. The parts 3 (Address Space Model) and Part 4 (Services) describe how to model and access to information, is important to understand because are used in the design of applications.
The Part 4 (Services) represents the relation and iteration between OPC UA Client and OPC UA Server, they define the information to be exchanged between

applications. The Part 6 (Service Mapping) defines the mapping of services to messages, security mechanisms and wire transport of message [13].
The information Model in Part 5 defines a framework for the information models in OPC UA, defines:

- The entry points into the address space used by clients to navigate through the instances and types of an OPC UA server.
- The base types building the root for the different type hierarchies.
- The built-in but extensible types like object types and data types.
- The Server Object providing capability and diagnostic information [16].

The Part 7 (profiles) defines the useful subset of OPC UA features implemented by UA Application to ensure interoperability between the subsets. The supported used profiles are exchanged in the establishment of connection between server and client and allow determining if the features needed are supported by the Communication partner.

The Part 8 (Data Access) defines how to use and represent the automation data and specify engineering units, The Part 9 (Alarms and Conditions) monitor the state of machines and the events. The Part 10 (Programs) define a state machine for execution, manipulation and monitor of programs while the part 11 specifies the use of historical access.

The two last parts Discovery and aggregates, describes how is the process to be discovered by a Client and a Server and how establish a communication.

### 2.3.6. OPC UA TECHNOLOGY

The technology that OPC UA implements for providing and ensuring interoperability between OPC UA Applications are: Coding, Securing the Channel and Transporting the data.

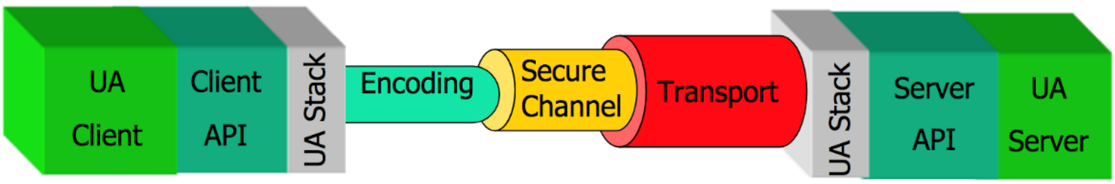



**Fig. 2.6** OPC UA Technology [17]

#### *2.3.6.1. Coding*

Depending on the network throughput OPC UA determine the optimal selection of encoding type, XML (Text Format) and UA Binary type. For data transmission XML format can be used and structured in different ways in the server depending of the data, if the throughput is large UI Binary format provide real time data and is used for saving performance providing fast encoding [12].

For exchanging data between applications or platforms the structures XML format is used. XML format can be parsed with a parser by applications. Applications in the enterprise level (ERP, MES) exchange data with operations level using XML because this format can deal with different types of data and is better for transport.

#### *2.3.6.2. Secure channel*

When the information is coded, it is automatically sent to the following layer the secure, which is in charge of keeping secure the information and complete. The secure channel uses digital signatures, authentication and authorization to keep safe the channel and use socket connection in a secure way.

There are two Security Protocols that works with certificates: WS-Secure Conversation and UA-Secure Conversation both work according to certificate connection [12].

#### *2.3.6.3. Transport*

The method of communication in OPC UA is message mechanism. OPC UA defined two transport protocols: UA TCP and SOAP/HTTP, both used in network level to make a connection between an OPC UA Server and Client. The first protocol accepted is UA TCP to stablish a reliable communication. This protocol have a message head and message body, the message body have two types of information, first the socket connecting information to build the protocol and second is the encrypted information transported [12].

The Second protocol accepted SOAP/HTTP is widely accepted by web service since is simple and firewall friendly. HTTP uses TCP protocol, SOAP combines HTTP technology and XML documents so is implemented to achieve interoperability between different applications and platforms, is flexible to exchange information.

### 2.3.7. Integration of a new device.

OPC UA is an open communication standard that due to its characteristics was designed to be implemented in interconnected environments such as Industry 4.0. There are many products certified by OPC UA [18] that work directly with the protocol and give the advantage of a direct integration, but there are also APIS and SDKs with free license and paid versions that allow developers to integrate their applications with this communication protocol. It leaves open a wide range of possibilities to be able to integrate applications to these interconnected environments.

For hardware developers an option is to use SDKs and libraries available if they are using tools such as a raspberry for a specific application. A raspberry is a tiny credit card size computer. To implement in Hardware that is going to be commercialized in the market the recommendation of OPC UA foundation is to make a certification and test of the products to be developed to be able to guarantee compatibility, interoperability, robustness and efficiency in the resources used. The certification process is carried out under the jurisdiction of Working Groups of OPC UA Foundation who establish all the rules and procedures. The advantages of the equipment certified by OPC UA include a fast initial configuration of the products, reliability and interoperability and minimum risks in the integration process. Certification tests last from 3 to 7 days and cost $ 950 per day for members.

More information about the certification processes are available on the opcua foundation website: https://opcfoundation.org/certification/overview-benefits/

## 2.4. NFC

NFC (Near Field Communication) is a wireless technology based in a short range standard that allows a simple, intuitive and bidirectional interaction between electronic devices. The goal of this protocol between terminals is without much effort configure a network which made easy and convenient for applications, only is necessary to bring them closer and is ready to use.

NFC works in an global frequency that not need licence (13.56Mhz) and a Bandwidth of 2Mhz, allowing the exchange of information between devices in a range of 10cm. This technology is based in RFID (Radio Frequency Identification) that was developed for communication with Automatic Identification Systems.

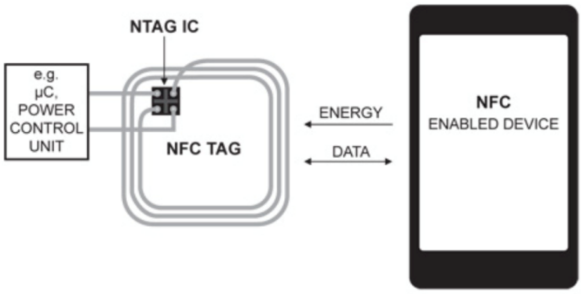



**Fig. 2.7** NFC Tag Working Principle [19]

The operating principle is using Magnetic Induction Fields as a mean of communication between two devices. The implementation of NFC in smartphones has allowed the development of many applications to improve the daily live.

### 2.4.1. Communication Modes

There are two modes of communication for this protocol.
Passive Communication Mode: In this mode the emitting device send RF energy to power the target device, the target modulates the energy in order to send back

a reply to the emitting device. An example of this mode is the communication between a mobile phone and a NFC tag.

Active Communication Mode: In this mode the emitting device and the receiving device have their own power supplies, one of them must deactivate its magnetic field to receive the response of the other device, so can work one device at a given time. The advantage of Active Communication Mode is that the data rate is higher and can work in longer distances.

### 2.4.2. Operating Modes

NFC Devices can operate in three different modes:
Mode Point to Point: This is the classical mode and allows the connection of data at a speed of 424kBit/seg this protocol standard is ISO 18092 and ECMA320/340

Mode Reading/Writing: In this mode NFC devices have the ability to read and write NFC Tags, the data transmission rate is 106kBit/seg.

Mode Emulate NFC-Tag: In this mode the NFC devices can emulate a NFC Tag with standard ISO 14443. In this mode the terminal phones can emulate a NFC Card and that is a Big Advantage for example in the use of credit cards

### 2.4.3. Advantages of NFC Technology

NFC Technology presents some advantages for its use:

- Easy to use, because is intuitive, only need to put together the devices.
- Can be used in different environments (industrial or domestic).
- Its working Operating Mode is regulated by Standards ISO, ECMA, ETSI.
- They are secure, because is necessary that the devices very close so to work.

### 2.4.4. Uses of NFC

NFC is having a great acceptance and have many applications that make it convenient for its use.

- Contactless payments: With this functionality the customers only need to use the phone and put it together to the NFC reader to make the payment, eliminating the use of credit cards and have effective.
- Share Information: The Small Tags allows to use it anywhere and may contain important information that can be accessed by bringing the phone closer.
- Public Transportation: Some Countries are implementing the use of NFC tickets for use in transportation.
- Health Area: In hospitals are used to follow and makes notes in the patients records in real time avoiding the use of printed paper.

- Replacement of the barcode: NFC Tag allows to replace the use of the bar code and have more information recorded in the NFC tags which can be tracked and read by NFC Readers.

## 2.5. ODOO

ODOO is a popular and powerful ERP software Open Source used by many small and medium companies, with more than 2 million users and in around 120 countries. This software is formed by modules that allows to manage different aspects from sales, financing, contacts, inventory and many others, all of them work interconnected and exchanging information.

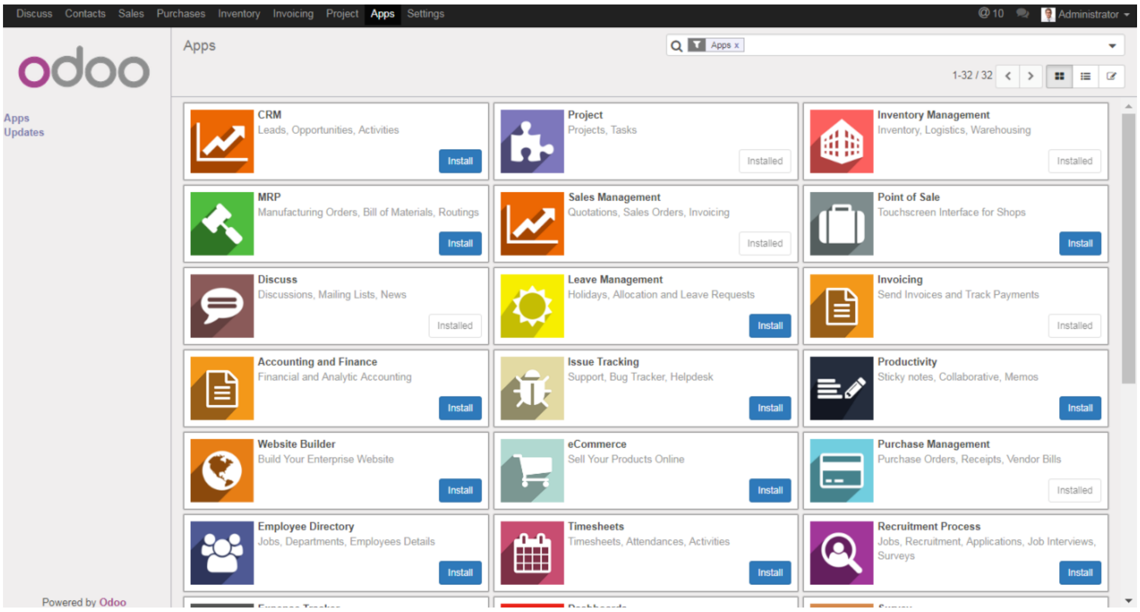



**Fig. 2.8** ODOO Software

ODOO works in parallel with a PostgreSQL database, in which the necessary records are stored. A great tool available by the community of ODOO is the Web API that allows access (read and write) in the registers and thus be able to integrate ODOO with other tools or external applications making it very powerful.

Among the available Modules the Inventory Management module is important because allows to have control of all the stocks available in the different warehouses, shelves or sections and to know where are the inventory, this module is of interest for this Thesis Project.

# CHAPTER 3. CONCEPT DEVELOPMENT

The idea of the Industry 4.0 is to have an interconnected environment, supported by the use of technologies as mentioned (Bigdata, IoT, Internet of Services, etc) achieving smart factories and intelligent places that will adapt to the current needs to improve and customize the production process optimizing the use of resources available and improving the Administrative Systems and the labour behind this due to the real time tracking of all the elements involved in the production.

This concept of interconnection between physical and digital systems in a company to have better control, optimization of processes and improvement in decision making, has been handled for a long time by large multinational companies. They invest large amounts of resources (money and people) in the research and development of **Ad-Hoc** solutions for its companies because this depends on the operation and coordination especially in logistics activities, where the response times in activities developed in a global scope are a key factor. An example of these companies are DHL, FEDEX, NESTLE, which continue to improve their work to achieve a great service to their customers.

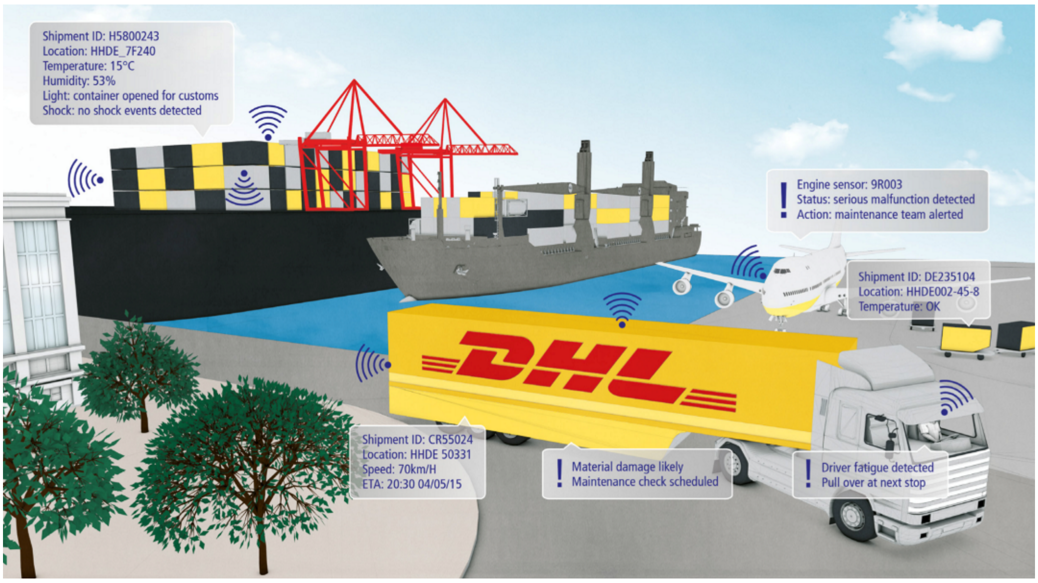



**Fig. 3.1** DHL Trends in Logistics[20]

## 3.1. Democratization of technology.

When a new technology or a new concept is introduced, the main objective should be to serve in the common benefit of the people that have small business or companies and that their monetary and technological resources are limited therefore don't have the time or the resources to develop an Ad hoc solution.

The benefits that are expected of the Industry 4.0 with its standard protocol OPC UA (open source) are that they are easy to implement for small and medium companies, at an affordable price and in relative short time, thus providing an accessible and democratically integration solution for the industries.

To enter in the Industry 4.0, it has been chosen to work in one of the areas with the greatest impact on the development of commercial, production and management activities in small and medium-sized companies, that is *Logistics*. The Logistics is a very important and valuable component in the movement of goods and raw materials between different places and the value chain of a Company, although there are technologies used to manage this, there are still many problems for its management and control, such as the lack of visibility of the goods transported along its entire chain supply, which causes uncertainty and errors controlling the movement between different places.

There are some manual tasks that are performed to control inventories, it is called the physical inventory, which is a control process that requires a physical count of all the pieces that make up the inventory. It is done once a year and it is a slow process and expensive, requires that the places or warehouses are closed for a whole day in order to count the items manually.

**Fig. 3.2** Manual Inventory Process [21]

In order to solve these problems is necessary to implement interconnected environments in an easy, accessible and integrable way that allows to locate in real time the location of the elements that are part of the inventory along the value chain, in this way there will be records and the whole route of movements of the products that can be verified and analyzed later.

## 3.2. Proposed Solution

Odoo is a suite of management (ERP) with modular applications adaptable and customizable according to the needs of each company and customer. It has several default modules (Purchases, Stores, Sales, Accounting, Inventories, etc.), and it is open source offering many advantages and possibilities of configuration. Currently it is widely used by small and medium enterprises for administrative and inventory control. The problem that exists in many companies is the manual task of inventory control and movement of things from one place to another. Since this is a manual task, there are many errors that appear throughout the control and require a long time to have this system updated. The proposed solution for this problem is that ODOO have a communication with the factory floor (vertical integration) so has implemented sensors or some way to automatically know the entire route of the inventory through the different points, knowing its route in real time.

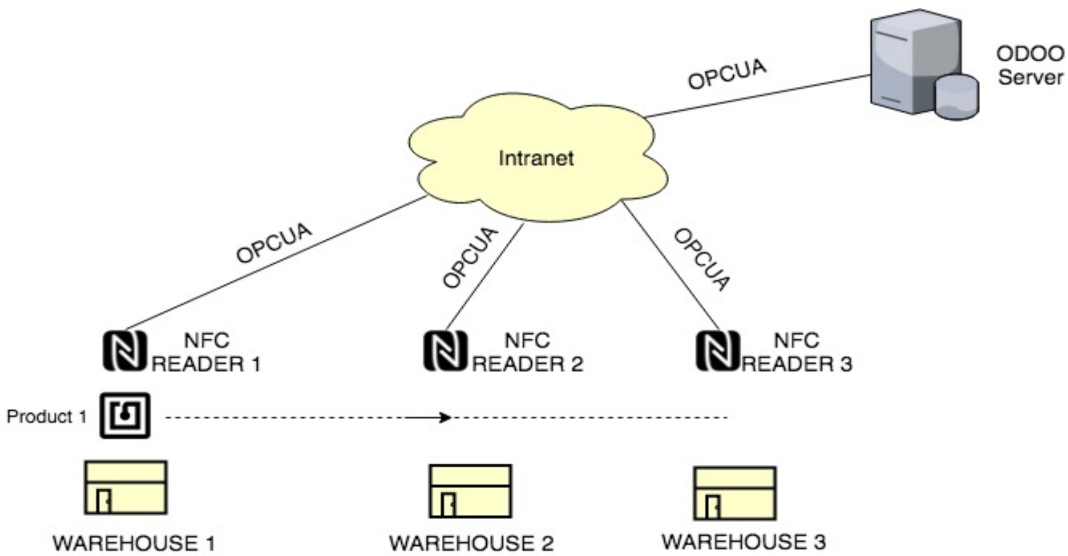



**Fig. 3.3** Proposed solution for Inventory Update

It is proposed to carry out a proof of concept of vertical integration between the ODOO software and identification elements (NFC TAGS) that need to be attached to every element that is part of the inventory. While an element is moving through the different sections of the warehouse, between different warehouses or between the supply chain, the employees will scan the NFC TAGS using a NFC reader (Raspberry Pi connected with a NFC Reader) and this information will be automatically updated in the ODOO Server, avoiding the need to update this information later manually.

The standard protocol for Industry 4.0 OPC UA will be used for the communication with all the advantages already mentioned before that this protocol offer. The implementation of the proof with this protocol opens the possibility of integrating any equipment that support this communication.

Each reader will send the identification code of each product plus the location code of the warehouse. This data will be used to update the records of the products in the ODOO database.

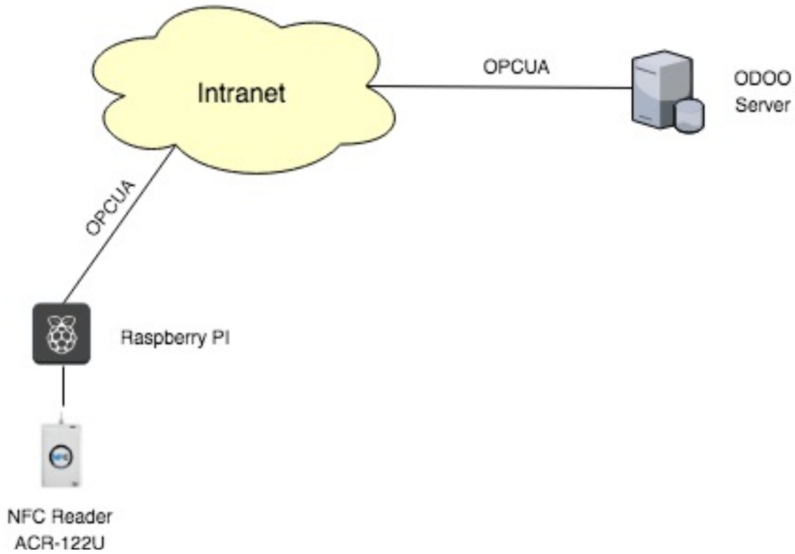



**Fig. 3.4** Scheme of the proof of concept

In the following section will be described the design and the implementation of the different parts that integrate this proof of concept.

## 3.3. Design of the proof of concept

The design of the proof of concept was carried out with the idea of providing a cost effective and easy to integrate solution.

The prototype to be developed is divided in the following parts:

- The tags or labels to be stuck in each element of the inventory to be tracked.
- The reader or the scanner that is an element to be located in different places of the warehouse and scan all the tags of the elements that pass by that point.
- ODOO Server that is in charge of receiving the location information sent by the reader and updating it in the ODOO Software.

### 3.3.1. Tags.

To be able to read and detect every element that is part of the inventory is necessary to use some type of tags to be stuck on each element, but each tag should not be expensive and easy to configure to record inside the EAN code that belongs to it and that will be detected later by the reader.

In between the different possible options are RFID tags and NFC tags, doing some research was decided that the most suitable in terms of easy to configure, size, availability and price are the NFC tags.

The NFC Tags are small integrated circuits incorporated with an antenna and that generally are integrated in stickers, actives or small cards. The advantage of using the NFC Tags are that its information is easy to access and execute using NFC devices as smart phones, tablets or NFC readers.


**Fig. 3.5** NFC Tags

The NFC-tags comply with NFC Forum type 2, are under the specification ISO/IEC 14443-3A, their price is low and allow to Read and Write.

The size of NFC tags are between 12 mm to 10 cm and the common used in the market are around 3 to 4 cm. The tags are very similar to a digital memory but with a lower capacity and don't require the use of external power supply because it works with electromagnetic field that comes from the NFC device and can be rewritable any times and protect the information stored with a password to avoid it of being deleted or modified.

In the market are available different types depending on the capacity and its memory configuration. In the following table we can see a summary of the models.

**Table 3.1. Types of NFC TAG**

| | **MIFARE Ultralight** | **NTAG 203** | **NTAG 210** | **NTAG 213** | **NTAG 215** | **NTAG 216** |
|---|---|---|---|---|---|---|
| **Memory Capacity** | 64 bytes | 168 bytes | 80 bytes | 180 bytes | 540 bytes | 924 bytes |
| **Memory Available** | 48 bytes | 144 bytes | 48 bytes | 144 bytes | 504 bytes | 888 bytes |
| **Character length** | 41 | 132 | 41 | 132 | 492 | 854 |
| **Mobil Compatibility** | Yes | Yes | Yes | Yes | Yes | Yes |
| **NFC Forum T5** | Yes | Yes | Yes | Yes | Yes | Yes |
| **Serie Number** | 7 bytes | 7 bytes | 7 bytes | 7 bytes | 7 bytes | 7 bytes |
| **Cryptography** | No | No | No | No | No | No |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Scan sensitivity** | Medium | High | High | Very High | High | High |

For the proof of concept is necessary to record the EAN (European Articule Number) Code of the product, the last version EAN-13 contains 13 characters and as it can be seen from the table all the models supports more than this length, so the criteria for choosing a model could be the price and size of the tags. For the tests have been used NTAG 213, this tags can be found in many sites and depending of the quantities can be found in a price of around $0.8 each.

### *3.3.1.1. NFC Tools. Writing a Tags*

The configuration of the tags is easy using an Android Mobile with NFC sensor incorporated and using the application NFC Tools to record. This application allows to Read NFC Tags, Write multiple type of data (Text, URL, Web links, etc) and perform some configuration as copy, delete, lock and format a tag.

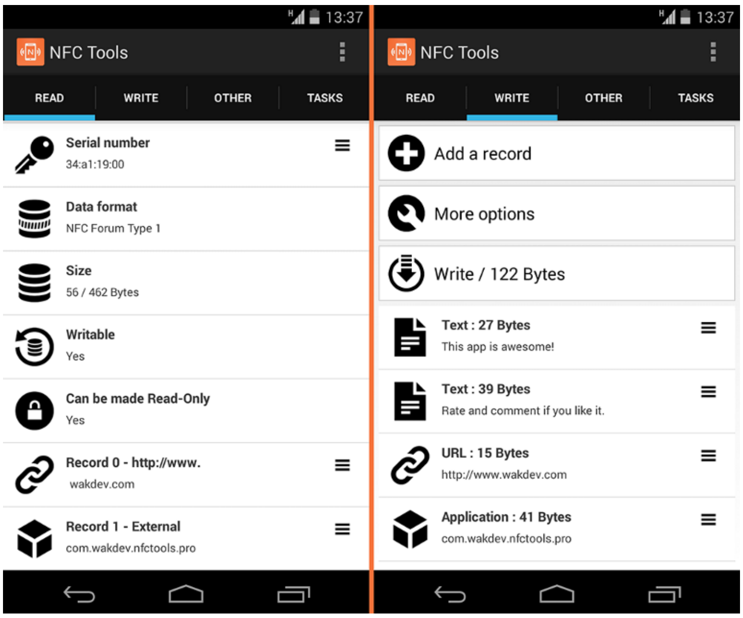



**Fig. 3.6** NFC Tools

As mentioned for the proof of concept we need to record the EAN-13 code, the procedure would be to type it manually in the phone application NFC Tools, record the code and then stick the tag in the product of the inventory.

### 3.3.2. NFC Reader

The reader should be a comfortable and inexpensive element that will be located in the different places of the warehouse and will scan all the products that pass by, thus registering their movement.

Here we have two possibilities of implementation. The first is using an android mobile phone with NFC reader support. In this case there should be a mobile application that allows to read the NFC tags, capture the internal location of the warehouse (the location can be a code that identifies the warehouse, the section or shelf) and then send it to the NFC server using the OPCUA protocol. The advantage of using this option is that is possible to use the personal mobile phone, just installing the application and with no extra hardware required.

The limitation that exists when implementing this option is obtaining an SDK (Software Development Kit) that allows executing an OPC UA server in the mobile terminal in order to send the information. In the tests carried out was tried to implement the OPC UA server in the mobile phone first using python code (using the Qpython application) but the python engine used was an Emulator, so has some limitations when updating or upgrading the libraries, what did not allow to execute the code. It was also tried to implement the OPC UA server in the phone using an SDK for Android Studio, It was requested a trial version of PROSYSOPC and there was found that for the moment it is possible only the implementation of OPCUA clients in the phone, that means that is possible only to read information and not to write so being discarded this option.

The second option that allows greater flexibility is the use of a raspberry pi with an NFC reader attached to it. This is very practical because it let to work with python code without restrictions and there is a very functional and documented library called FreeOPCUA in python 3 that allows to implement client and server OPC UA. The implementation could be a basic communication or a communication with the use of certificates and keys that provide a layer of authentication and security, In this case as it is a proof of concept it is not necessary to use certificates and keys for the demonstration.

Once the issue of OPC UA protocol communication has been resolved, it is necessary to use an external NFC reader to be connected to the raspberry and allow the scanning of the tags. To be able to use the NFC reader it is necessary to acquire a reader, install the different drivers for its operation and install the library that allows its control from python. Before buying we must verify the support of that Reader model in python checking in the web site of the NFC library.

The model chosen for this prototype in terms of physical form, functionality and price is the model ACR ACR122U USB2.0. This reader support MIFARE®, ISO 14443 A/ B, NFC & FeliCa, works using USB and have the support to work in linux with some github libraries.

The following table shows the features of the reader.

**Table 3.2. Features of ACR122U NFC Reader**

| | |
|---|---|
| **Dimensions** | 98mm x 65mm x 12.8mm |
| **Weight** | 70g |
| **Interface** | USB 2.0 |
| **Compliance** | CCID |
| **Reader** | Read/write speed up to 424 kbps |
| **Reading Distance** | Up to 50 mm |
| **Support** | SO 14443 Type A and B cards, MIFARE, FeliCa, and all 4 types of NFC (ISO/IEC 18092) tags |
| **Programming Interface** | PC/CS, CT-API |

In the following image is possible to see the physical appearance of the reader that is going to be used.


**Fig. 3.7** ACR122U NFC Reader

This reader works with a library called "smartcard library" and is supported only by python 2. The reader system should read the NTAG and send this information through OPC UA protocol so is necessary to have an script doing this task. Due that the NFC reader works with python 2 and the OPC UA library works with python 3, that means that are incompatible to work together, is necessary to implement 2 scripts, one for reading and one for sending the information, and between them they can communicate using an UDP socket. In the following Figure is possible to see the Diagram of Scripts used in the reader.

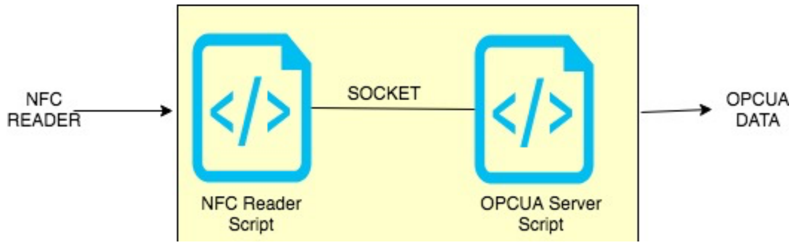



**Fig. 3.8** Diagram of scripts in the reader

For the system to work, it is necessary that the two scripts are working at the time simultaneously.

The information sent from the raspberry to the ODOO server is composed as follows: Warehouse code + EAN
In this way, depending on where the NFC reader is located, the warehouse code will change and the ODOO server is in charge of updating the data of the inventory module.

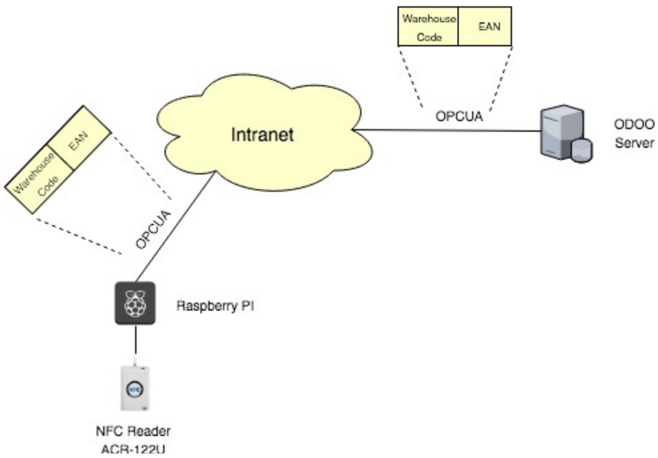



**Fig. 3.9** Structure of the information sent in the protocol.

The different warehouse locations where the products will be stored must be manually entered into ODOO using the warehouse management menu. ODOO automatically assigns them an internal product location code, which can be consulted using the Web API and the code used is appended.

In the following table some codes assigned by ODOO to some locations are shown:

**Table 3.3. Warehouse codes.**

| Warehouse Code | Place |
|---|---|
| 9 | Partner Locations/Customers |
| 12 | WH/Stock |
| 18 | WH/Stock/Shelf 2 |
| 19 | WH/Stock/Shelf 1 |
| 27 | Chic/Stock |

### 3.3.3. ODOO Server.

As previously mentioned, ODOO is an open source ERP that contains many modules of importance for the management of companies, among these modules it is of especial interest the Inventory Module because allows to have control of all the stocks available in the different warehouses, shelves or sections and to know where are the inventory. The problem is that this module require to do manually the register of moving of the items from one place to other, what is pretended to do is to update automatically the location of the products when they are moving between different places inside of the company without the need for a manual intervention to keep the location updated.

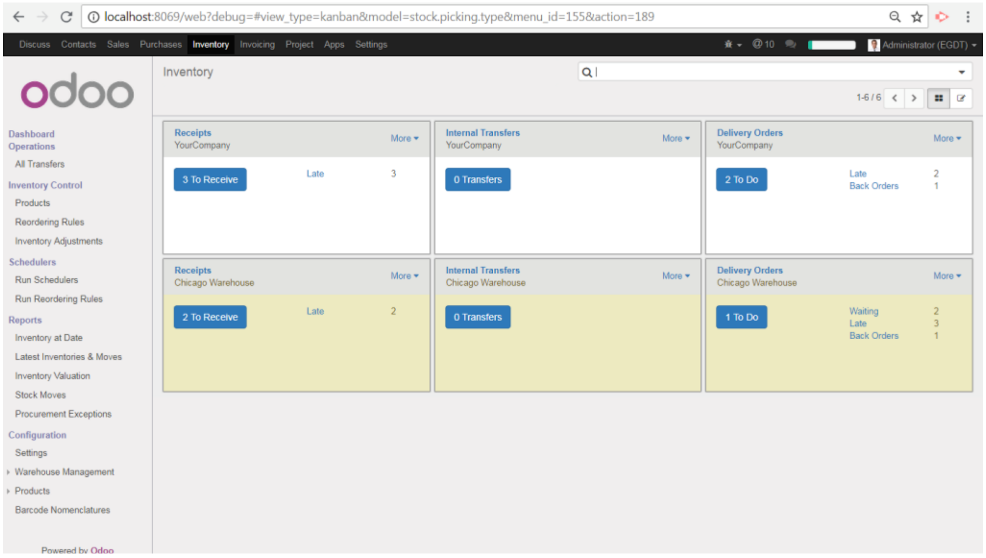



**Fig. 3.10** Inventory Module in ODOO

For this part is necessary to have installed in one computer the ODOO Server 9.0 which will be running all the time and providing the access to the employees. The advantage is that this software is free and open source so can be downloaded and installed in any operative system from its web site: https://www.odoo.com/es_ES/page/download. For this Master thesis will be used Windows 7 and the ODOO version 9, compatible with it.
After having running the ODOO Server is needed to have a OPC UA Client listening for incoming information (EAN Code + Warehouse Code) and storing this in the ODOO Server.

In order to have access to ODOO and its registries it is possible to make use of the Web Service API which allows to create new records, consult and modify them. This API is available for use in python 3 and to be able to access to ODOO is needed to know the URL of the ODOO server, its access port, the name of the database, the user name and the password.
Due that OPC UA client and the ODOO API work in python 3, they both can be implemented in a single python script that will be running all the time waiting for

the arrival of the data to later using the EAN code locate the product in ODOO and update the new location in the server.

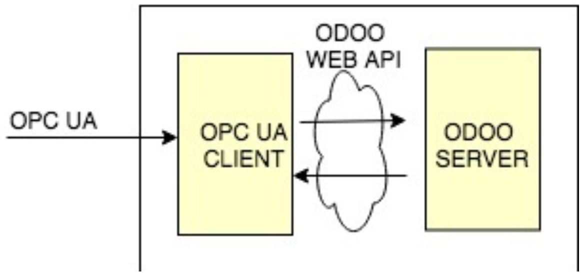



**Fig. 3.11** OPC UA Client with ODOO Server.

## 3.4. Implementation of the proof of concept

The prototype is implemented following the diagram of the figure 3.4 and is composed of the following parts:

- NFC Reader capable of reading NFC Tags with NDEF information stored inside.
- A Raspberry PI with a python script that will read NFC Information and other python script that will run an OPCUA Server to send the information.
- A Windows virtual machine that will run an ODOO server and a python script running a OPCUA Client and with ODOO API.

The operating process of the system is described below:

1. All the items belonging to the inventory must have an NFC tag attached to the product and an EAN identification code will be recorded inside to identify it. This same code must be recorded in the barcode field in ODOO which can be located in the General Information tab of the product.
2. To record the EAN code in each item of the inventory, we use the android application NFC Tools, which among its options allows the writing and protection of the information with a password so it cannot be modified.
3. The NFC Reader will be located at some point of the warehouse, and each time that an item arrives or circulates by this point (it can be a shelf, a warehouse, an office, etc) it should be scanned by the reader, the raspberry will process in python the detected code and stablish a communication through the OPCUA protocol with the OPC UA Client with ODOO Server.
4. At another point will be located the OPC UA Client with ODOO Server, that will have an script in python listening for OPCUA communications and receiving the product codes with its new location ID.
5. Once the ODOO server receives some item code and the location, using the ODOO API will check for that product in ODOO and will perform an update of the location code keeping the data up to date.

The flowchart of the Reader and the ODOO server is shown in the next figure.

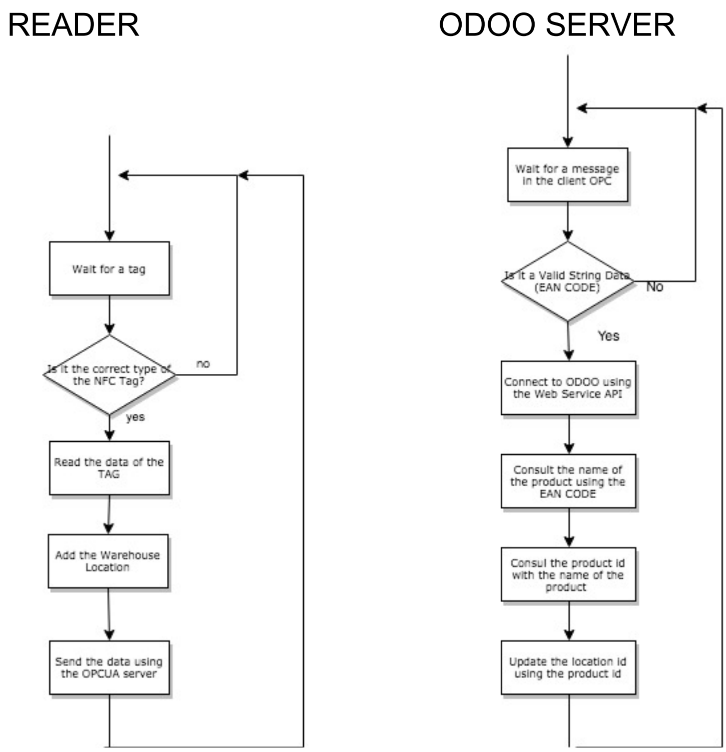



**Fig. 3.12** System Operating Flow Diagram.

### 3.4.1. Integration of NFC Reader with OPC UA Server

The reader is composed by an NFC Reader model ACR-122U connected via USB to a Raspberry PI that work as a OPC UA Server and is responsible for sending the data to the ODOO Server.

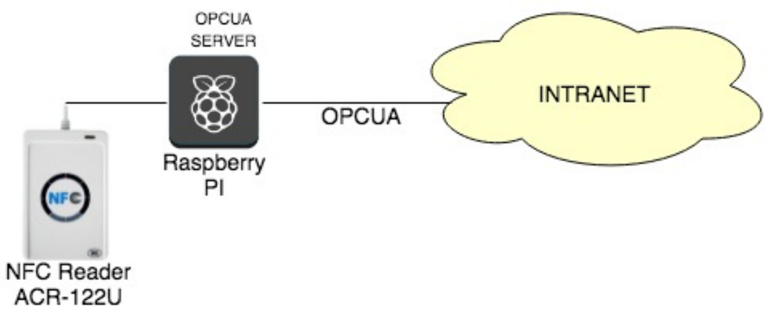



**Fig. 3.13** NFC Reader composition.

To be able to read the NFC Tags in the raspberry is necessary:

1. Install the python Library called “smartcard library” available at: https://sourceforge.net/projects/pyscard/files/
2. Install the drivers for the reader available at :

https://www.acs.com.hk/en/driver/3/acr122u-usb-nfc-reader/

3. Clone the library available in the raspberry from the terminal:
   git clone git@github.com/pmoncadaisla/eurinfc

Inside of folder from step 3 (eurinfc) there is a file called read.py which already gives as an output the reading of the TAGS. It is important to have all the files running in the same folder because all of them form part of the same module.

**Fig. 3.14** NTAGs used for each item

To install the Free OPC UA library in the Raspberry Pi is necessary to have a Python 3.4 or greater and have installed the following from the terminal:

1. Install cryptography Library : pip3 install cryptography
2. Install dateutil: pip3 install dateutil
3. Install lxml: pip3 install lxml
4. Install freeopcua: pip3 install freeopcua

Once it is installed in the raspberry the NFC reader and the OPCUA protocol, it is necessary to develop the python script that will control the process. In the following figure we can see a block diagram of the structure of the control script.

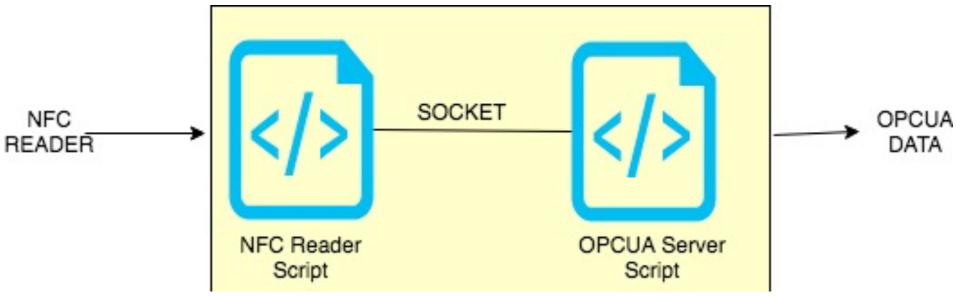



**Fig. 3.15** Structure of the software in the Raspberry

The NFC reader was developed to work with python 2 and the OPCUA protocol works with python 3 so it was necessary to develop two scripts, one for the NFC reader and another for the OPCUA communication and the communication between the two scripts was internally using a UDP socket.

In the Annexes we can check the code used for reading the tags and sending in a OPCUA server.

### 3.4.2. Integration of OPCUA Client with ODOO Server

For the proof of concept, a virtual machine of Windows 7 was installed in a MacBook Pro computer and was used the Parallels Software Virtual Machine to run the ODOO Server, the ODOO server and the OPCUA client were running on the same computer. First it is necessary to have installed the ODOO server which can be downloaded from its web site available for Windows and Linux version: https://www.odoo.com/page/download. Its installation process is simple since there is a wizard that guides you step by step in the installation. The installation of ODOO also involves the creation of a database in PostGreSQL.

Once the software has been successfully installed, a user as Administrator is created, from which it is possible to access and manage the internal configurations and the rights to make any changes as well as assigning new users and allows to install the different applications according to the needs of the business such as: sales, manufacturing, mass mail, invoices, etc. We have to make sure that the Inventory module is installed, which is what the final result will be reflected in.

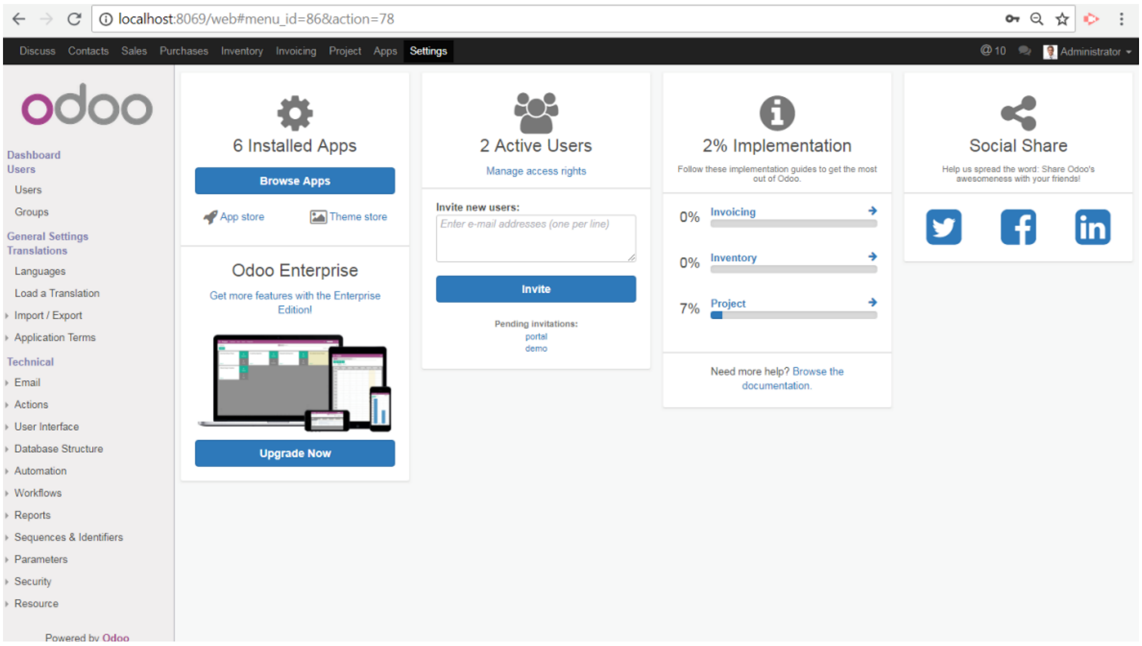



**Fig. 3.16** ODOO Administrator User

To have the OPC UA client listening is needed to have installed python 3 on the server machine and then the FreeOPCUA library with same required modules that were used for the server OPC UA, the steps for installing are:

1. Install cryptography Library : pip3 install cryptography.
2. Install dateutil: pip3 install dateutil.
3. Install lxml: pip3 install lxml.

4. Install freeopcua: pip3 install freeopcua.

The implementation of the ODOO Web Service API does not require the installation of an extra library, since the necessary xmlrpclib is already installed in python 3. Fist the connection is established using the API with the ODOO server then is made a query to determine the code of the internal product in ODOO using its EAN code and finally the warehouse location code is updated.

The programming code to generate the OPCUA client and the code to use the ODOO API are written in the same script and to perform the tests was used PyCharm which is a python IDE. In the Annexes section, the code lines used for its implementation are specified.

### 3.4.3. System Test and Analysis

To be able to carry out the test, it is necessary to have correctly installed and running our ODOO Server, in which the expected and planned results in the development of this master thesis will be reflected. To have access to ODOO server it is necessary to use a web browser and type the URL and the port of the server, then will be asked the database name, user name and password.


**Fig. 3.17** ODOO access window

For the correct working of the test is necessary to have activated the option that allow multiple locations in a warehouse in the inventory module as shown in the image.

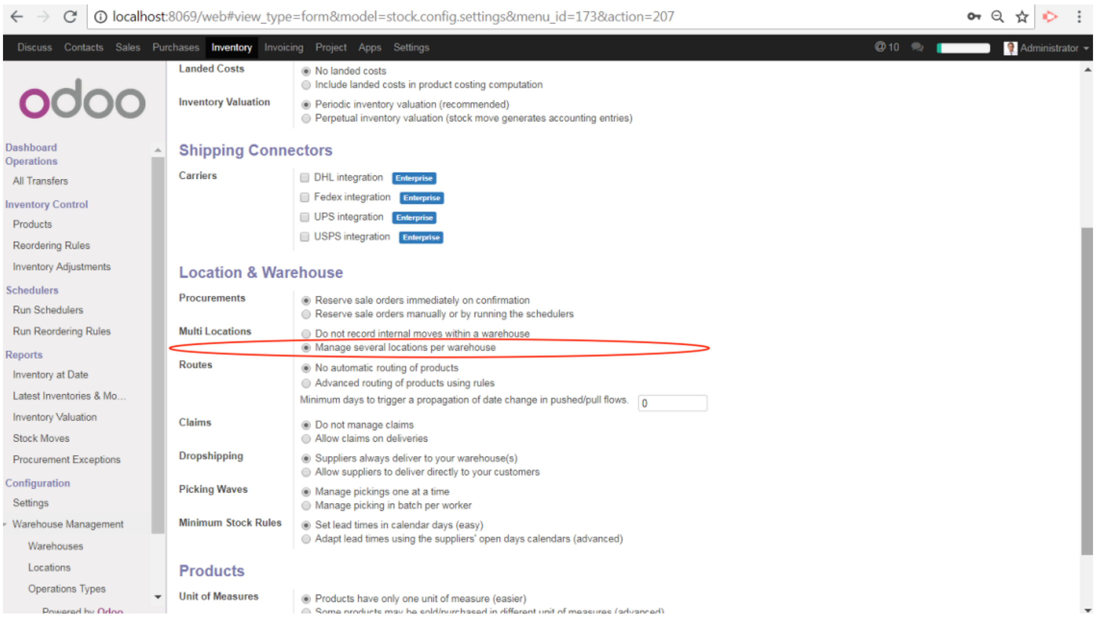



**Fig. 3.18** Several locations Configuration

Using the warehouse management menu it is possible to manage all the warehouses and internal locations where the products can move. In the graph is possible to see the existing locations that are created by default by the DEMO database.

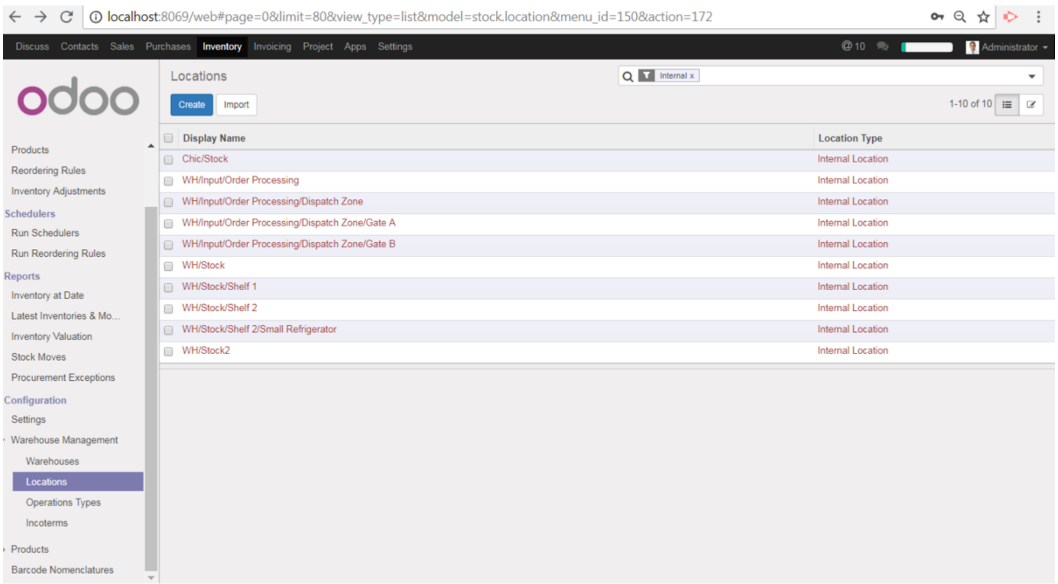



**Fig. 3.19** Warehouses and Locations in ODOO

ODOO allows to have two types of user access, as administrator with administrative privileges or as a user that can perform basic query tasks.

Now the user (with administrator privileges) can manage and configure the products that will have available in the inventory. In the products menu of the inventory module is possible to create or modify the products that are going to be handled as well as to register its correct EAN code that will identify it along the chain.

As an example has been created a product, a cell phone (Phone Huawei P8) for which the basic identification data as name, reference, price have been entered and in the Barcode field the EAN identification code has been placed.

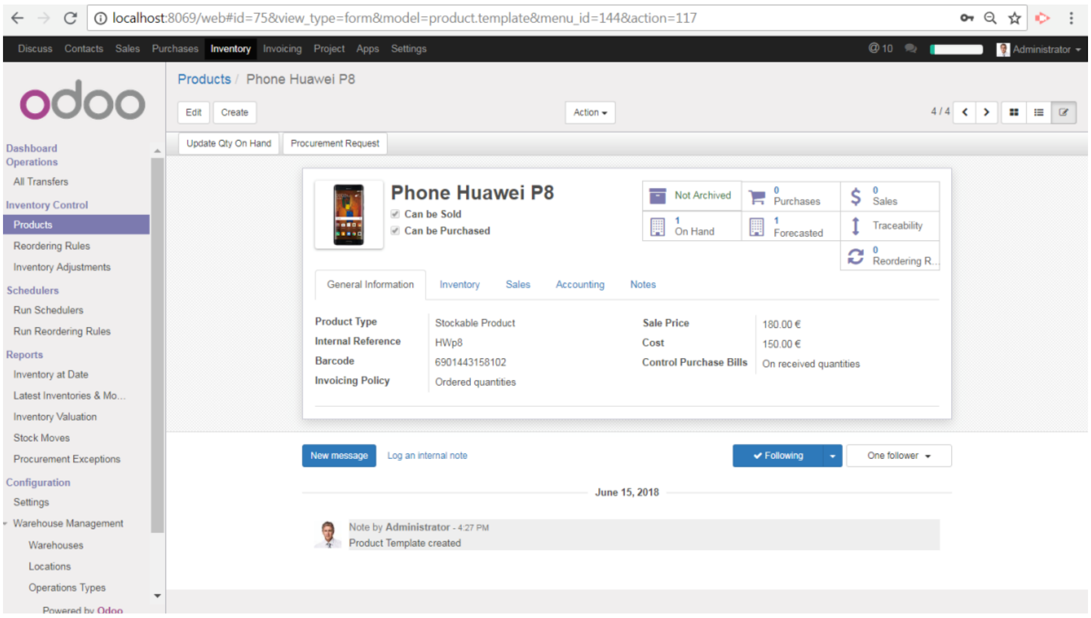



**Fig. 3.20** Configurations of a product in ODOO

By clicking on “Update Qty On Hand” option it is possible to update the initial place where a product is located at the moment of its creation.

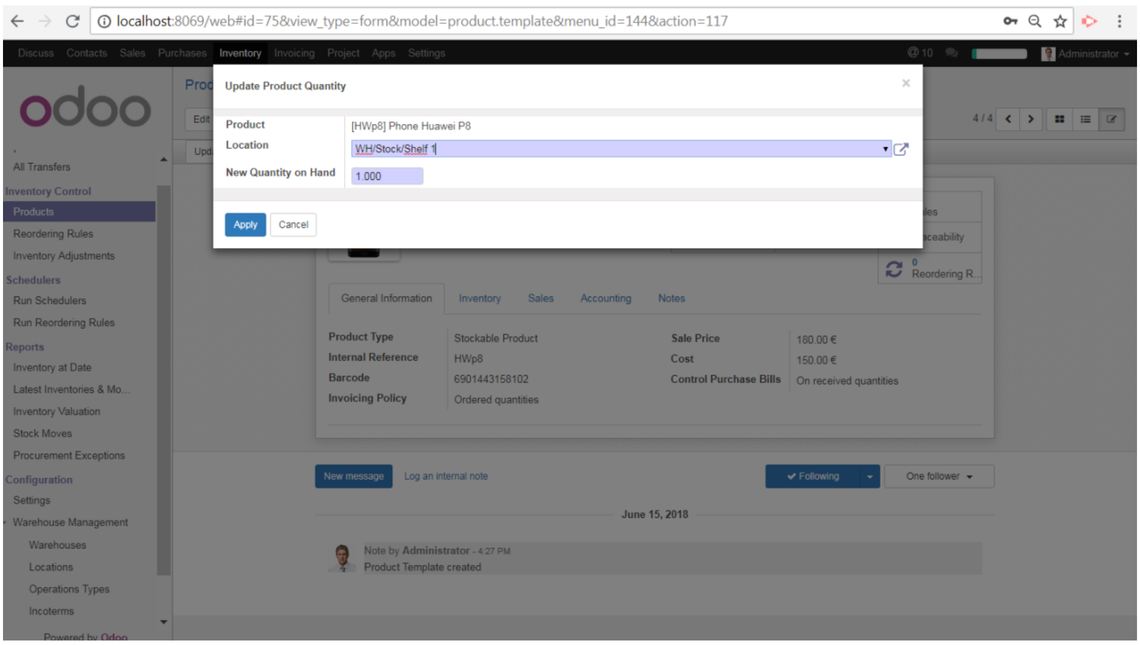



**Fig. 3.21** Update product initial location

Now it is possible to verify at any time the current registered location of the product by clicking on the “On Hand” button as shown in the following figure.

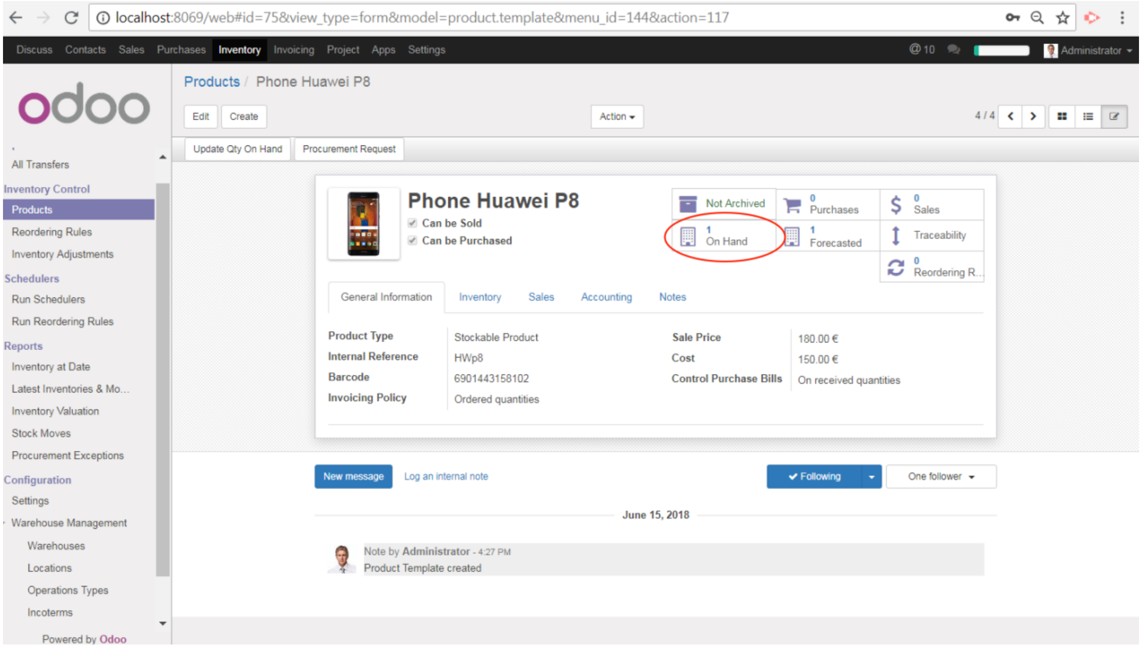



**Fig. 3.22** On Hand Bottom Location

In the following figure is verified the location of the product Phone Huawei P8 that is currently located in the Warehouse WH inside the section Stock in the Shelf 1.

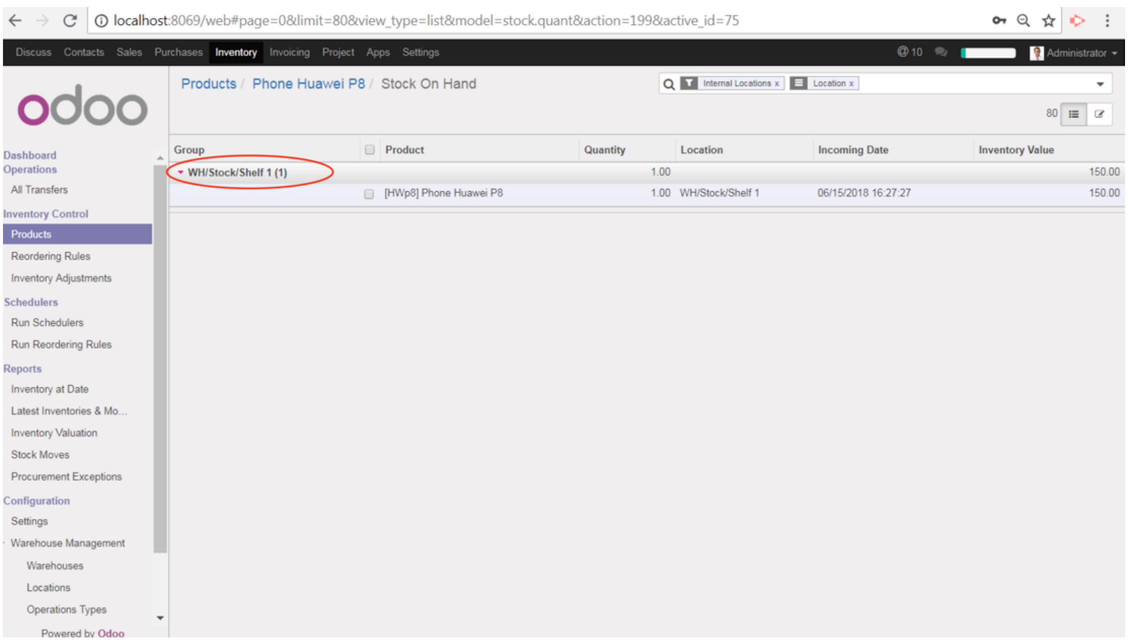



**Fig. 3.23** On Hand Product Information

The next step is to configure the tag with the information of the EAN code that corresponds to the selected product for which the android application NFC Tools has been used to record the information. The following image shows the EAN information stored in the tag.

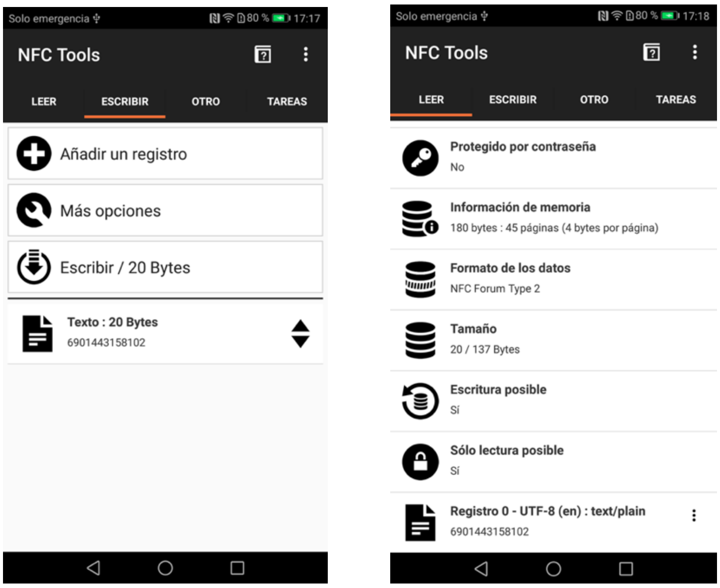


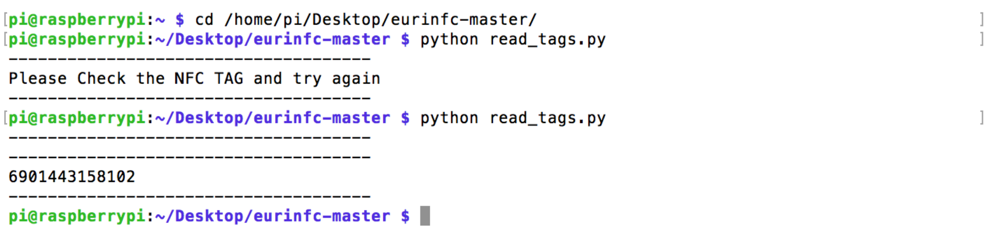

**Fig. 3.24** Writing (Left) and Reading (Right) screen of NFC tools

As seen in previous image, first was written the EAN code that belongs to the phone and later was verified with the same application using the reading function.

To start the reader (Raspberry and NFC Reader)  it is necessary to run two scripts, which can be done through the terminal by SSH from the computer.

What should be paid attention is the IP address of the OPC UA client and the IP address of the OPC UA server which must be updated in the scripts before executing them so that there are no problems or execution errors.

To read an NFC tag it must executed an script called read_tags.py as shown in the following graph:



```
pi@raspberrypi:~ $ cd /home/pi/Desktop/eurinfc-master/
pi@raspberrypi:~/Desktop/eurinfc-master $ python read_tags.py
-------------------------------------
Please Check the NFC TAG and try again
-------------------------------------
pi@raspberrypi:~/Desktop/eurinfc-master $ python read_tags.py
-------------------------------------
-------------------------------------
6901443158102
-------------------------------------
pi@raspberrypi:~/Desktop/eurinfc-master $
```

**Fig. 3.25** Execution of the script that read the tags

As seen in the picture is necessary to run the script and if  in case there is some error doing the reading a message will appear saying “Please Check the NFC TAG and try again” otherwise  will be printed the EAN code that was read.

In parallel, a script running the OPC UA server must be running, the script name is send_opc.py. This script is working all the time and the first time it runs takes a few seconds to start the server. In the following figure is possible to see this script running.

```
pi@raspberrypi:~/Desktop/eurinfc-master $ ^C
pi@raspberrypi:~/Desktop/eurinfc-master $ python3 send_opc.py
add_node: while adding node NumericNodeId(i=15957), requested parent node NumericNodeId(i=11715) does not e
xists
add_node: while adding node NumericNodeId(i=15958), requested parent node NumericNodeId(i=15957) does not e
xists
add_node: while adding node NumericNodeId(i=15959), requested parent node NumericNodeId(i=15957) does not e
xists
add_node: while adding node NumericNodeId(i=15960), requested parent node NumericNodeId(i=15957) does not e
xists
add_node: while adding node NumericNodeId(i=15961), requested parent node NumericNodeId(i=15957) does not e
xists
add_node: while adding node NumericNodeId(i=15962), requested parent node NumericNodeId(i=15957) does not e
xists
add_node: while adding node NumericNodeId(i=15963), requested parent node NumericNodeId(i=15957) does not e
xists
add_node: while adding node NumericNodeId(i=15964), requested parent node NumericNodeId(i=15957) does not e
xists
add_node: while adding node NumericNodeId(i=16134), requested parent node NumericNodeId(i=15957) does not e
xists
add_node: while adding node NumericNodeId(i=16135), requested parent node NumericNodeId(i=15957) does not e
xists
add_node: while adding node NumericNodeId(i=16136), requested parent node NumericNodeId(i=15957) does not e
xists
Listening on 192.168.1.102:4842
Server Started at
---------------------------
Sent to ODOO:
---------------------------
code:  6901443158102
location:  19
---------------------------
```

**Fig. 3.26** Execution of the script that run the OPCUA server in the reader.

The first lines that are shown are execution messages that appear when is starting the server, then every time that an NFC tag is read, the EAN code of the element and the warehouse location code will be printed on the screen and sent via OPC UA to the OPC UA Client with ODOO Server.

This script send all the data that receives from the server if the received data is in the correct NDEF format for NTAGs. This script also sends the warehouse location code that belongs to the location of the reader, if by case is moved, the new location code must be updated in this script.

Finally the script located in the virtual machine (OPC UA client with ODOO Server) must be executed. As was said before this virtual machine is working with windows 7 and was used PyCharm as a python IDE for testing. In the following image the script running from PyCharm is shown:

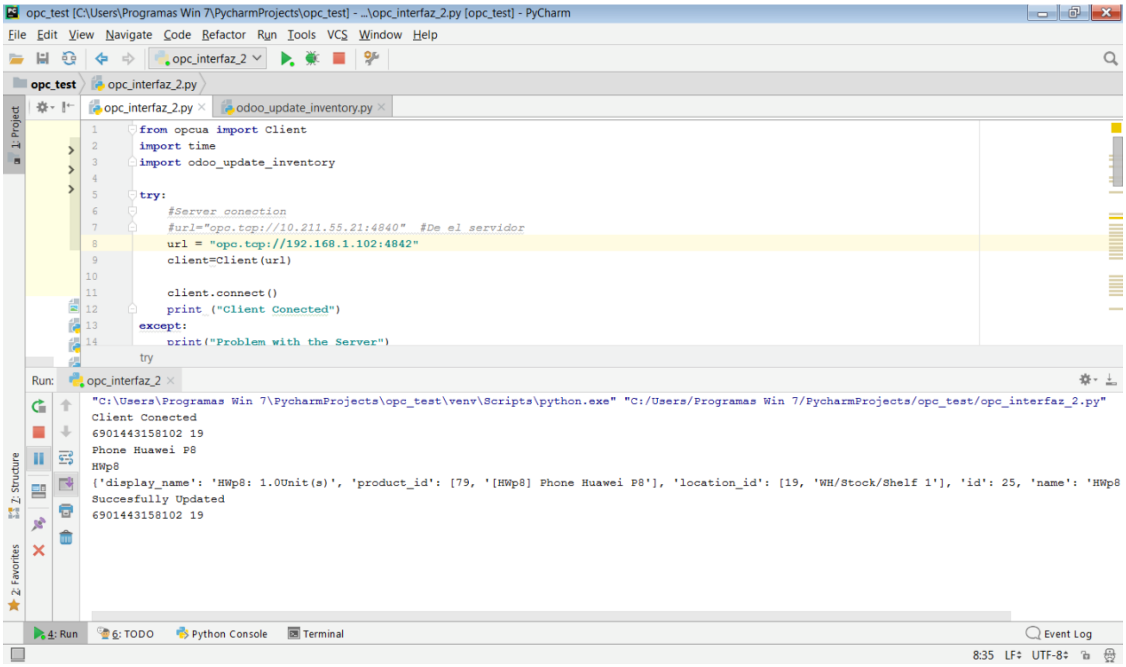



**Fig. 3.27** Execution of the script that run the OPCUA client

The Image shows that the data was received and updated according to the execution terminal. To verify if was updated correctly its necessary to access in ODOO and check the location of the item scanned as the image shows.

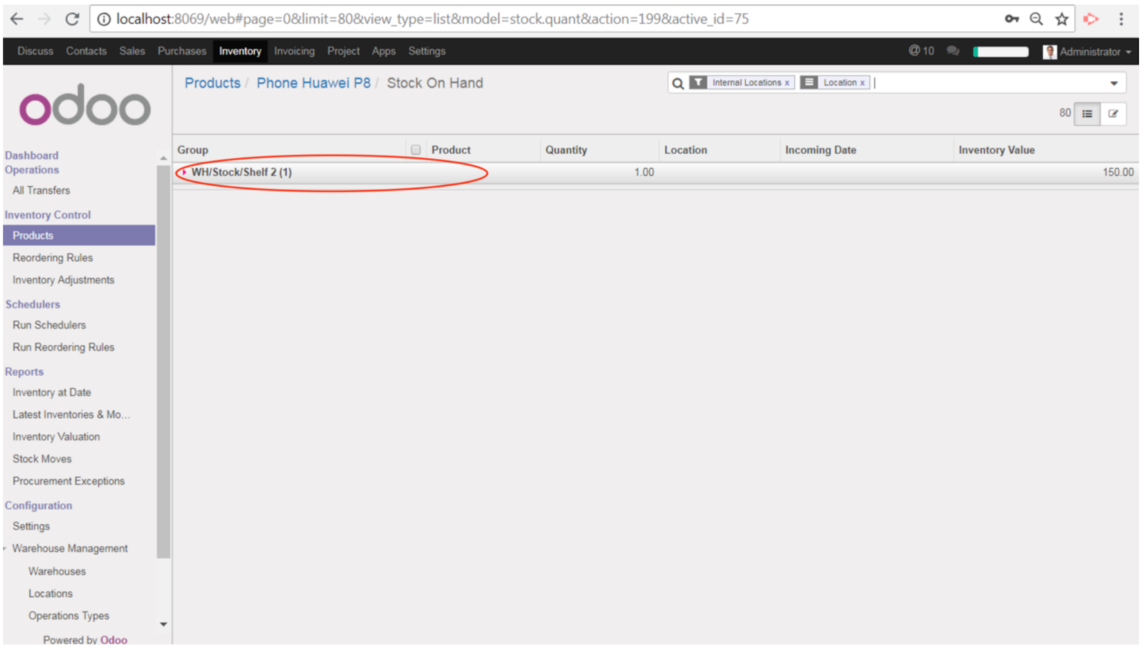



**Fig. 3.28** Execution of the script that run the OPCUA server in the reader

It is possible to check in the figure that the product has moved from the shelf one (figure 3.23) to the shelf two (figure 3.28).

## 3.5. Business Model Opportunities

The use of OPC UA as standard communication protocol for Industry 4.0 opens an incredible possibility of interconnection between different equipment and at the same time the possibility of sharing important information between different companies that will allow dynamic interaction for all. The previous knowledge of this information will help to reduce time and increase productivity. There is a great motivation to share information between companies, because is the possibility of new business models.

For Example a company provider offers the automatic update service in inventory control in real time. This provider company can sell or lend the devices (NFC Readers) to its clients ensuring that there’s is a continuous information update of ODOO and charging for the provided service.

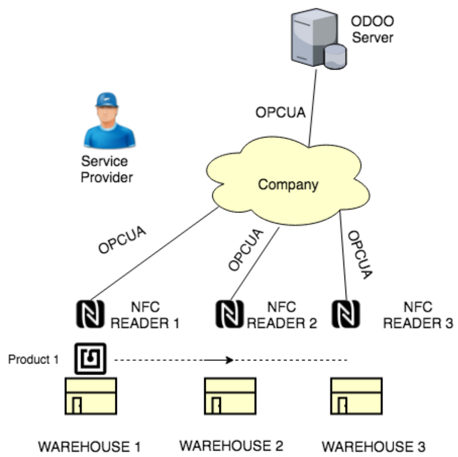



**Fig. 3.29** Inventory Control in a Company

This same provider could offer their inventory control update services to their client supplier and to their client’s clients, providing as a benefit the update of information in the clients ODOO and in the ODOO of the supplier, thus knowing the total displacement of the product from the Supplier Warehouse to the final destination.

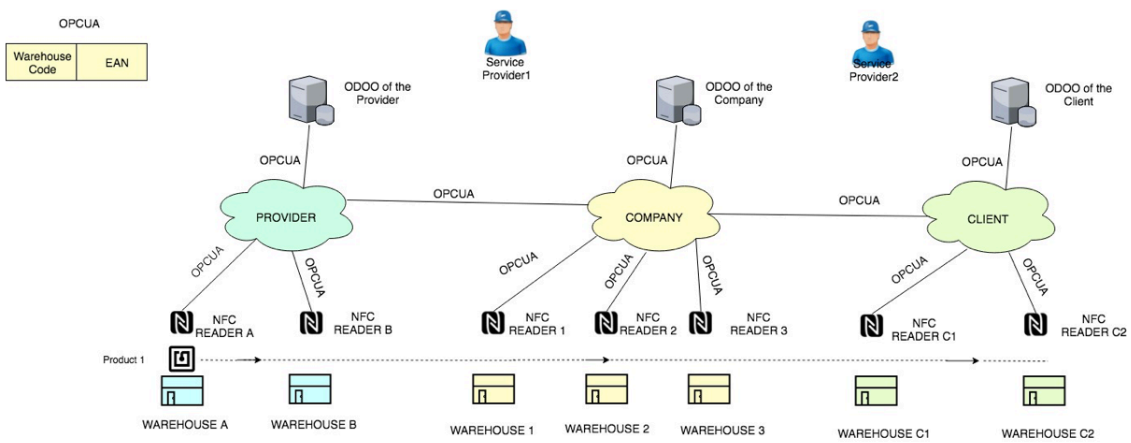



**Fig. 3.30** Information Shared to Provider, Company and Client

The Advantage of this service is that is not an AD-HOC solution, is a solution developed using simple tools and with an standard protocol OPC UA that guarantees the security of the information thus democratizing its use. Each NFC reader can feed with information to many ODOOs pointing to the IP Address of the client ODOO or the provider ODOO.

With this model using a free protocol like OPC UA is possible and feasible the existence of several providers that use the same protocol and different hardware and that are interoperable existing the possibility of competition and coexistence in the market, differentiated only by the provided service offered.

Solutions as this Control of inventories are a sample that the Industry 4.0 can be implemented to solve daily problems of a company and would be the starting point toward a digitalization. Once the first steps are given is possible to take or to share information in a collaborative environment.

# CHAPTER 4. CONCLUSIONS

## 4.1 Conclusions.

The development of this Master Thesis begin with a review of the Industry 4.0, its evolution and the technology that supports it, focusing on the OPCUA protocol that will allow the interconnection and compatibility of several machines, devices and services.

It was proposed a Challenge, the designs and development of a proof of concept that solves a problem of logistics present in medium and small companies, that is the control of inventories, using ODOO (ERP system and open source), the OPC UA protocol and NFC Readers. After Finishing the proof of concept, it can be said that:

- It was possible to perform a vertical integration with ODOO and the NFC Readers, since most of the works and research consulted carried out horizontal integrations between equipment or devices of the same level.
- For the implementation and use of the OPC UA protocol there are SDK´s and libraries available (free OPCUA, OPCUA foundation, Matrikon OPC, Prosys) some options are free and others are paid. Different options were tested and finally was decided to use freeOPCUA, this library have good examples of implementation.
- The Android Platform has a huge limitation with the implementation of the OPC UA protocol, because only allows to configure as a client and not as a server, which limits its use to reading and not writing information. Several tests were carried out on this platform but it was decided to discard it due to the non-achievement of the expected results.
- Due to the limitations present in the android platform for the use of the OPCUA protocol, the best implementation option for the NFC reader was a raspberry using python as a language.
- The inconveniences of using the NFC reader with the raspberry was finding the correct library and drivers for using them so was necessary to test two different types of reader (module PN532 and module ACR122U) selecting the second alternative because its drivers allowed the correct reading of the stored information in the tags.
- The writing and reading of information in the NFC tags is fast and simple using the free application NFC tools available only for android. NFC tags are a great alternative for economical and feasible labelling.
- ODOO Server 9 was installed and tested in a MAC Computer with Windows 7 Virtual Machine and its installation was simple, following the steps and recommendations given by the wizard.
- The ODOO Web Service API for external applications has a good documentation and allowed to access and update information in the ODOO server using python 3 script in the Windows 7 Virtual Machine with PyCharm as an IDE for windows platforms.
- The script that runs the OPCUA server take some seconds the first time is turned on, but after that the communication is immediate.

- Cell phones offer more portability for NFC reading applications so it is recommended for future projects to look for alternatives to implement OPCUA servers in android which will facilitate the use of the personal phone is not need for external hardware.

The proof of concept showed that is possible to democratize the technology and implement an Industry 4.0 for small and medium companies integrating accessible tools such as ODOO, raspberry, NFC tags and OPCUA protocol having a real time solution. It can be taken the advantage of the protocol and integrate other equipment's and systems to exchange information in real time having smart environment's.

From my point of view and after the experience gained in the development of the project I think that the implementation of Industry 4.0 in companies is feasible and necessary, without the need for large investments in Ad-hoc solutions, it will allow to optimize resources and have solutions in real time and will require specialized people in different technologies. Its integration will not be quick because digitizing the processes of a company in many cases requires a redesign of the entire system which compromises and imposes a barrier in the short term but its long-term advantages are really important. Although there are projects by governments of some countries to support with policies and support the digitalization of companies toward and Industry 4.0, for the small and medium enterprises it is a challenge that will take about 5 years to become aware of the advantages and contribution that these solutions bring to the Industry.

For future work could be done many things starting with what has been tested in this master thesis. It could be started by solving the same problem of inventory control proposed but this time integrate more NFC tag readers and implement more ODOO servers that would represent different companies and customers located along the value chain and share the information to be updated simultaneously between the different ODOOs. These changes will require a modification of the script that updates the information in the ODOO server and also the script of the NFC reader needs to stablish communication with all the ODOO servers to be updated, probably some challenges to be faced will appear when several servers and readers are working simultaneously but will not represent a big problem. Each client and each server has its own IP Address assigned which facilitates the identification and communication and allows them to be physically installed in different places and different networks and to stay connected. Once the above will be achieved, it would be interesting to integrate OPCUA certified equipment's into the system to obtain another type of sensor information that could be useful to feed more data in ODOO servers, demonstrating the great possibility of growth, interoperability and integration that the protocol offers. When the previous scenario is done, could be performed a performance tests between equipment certified by OPCUA and systems that works using libraries and SDKs. The fact that the system is working with an open protocol such as OPCUA leaves open the possibility of integrating everything in the same system sharing important information.

## 4.2 Sustainability Considerations.

The implementation and use of Industry 4.0 economically in the medium term involves a reduction of costs for the company which represents savings, this is because the use of resources is optimized thanks to the new technologies implemented, the social impact caused by Industry 4.0 It is more reflected in the employment, on the one hand there will be people who will be replaced in their work by a robot or computer but on the other hand requires the preparation in new areas of knowledge of people so they can contribute to new jobs. The environmental impact will not be greatly affected because with the Industry 4.0 is intended to optimize the use of resources which could achieve a saving in energy consumption.

## 4.3 Ethical Considerations

Implementing an Industry 4.0 in companies means that they will be efficient in productivity and management, which would generate an economic impact contributing to increase the profits and reducing cost. A lot of information will be shared between systems and this information can be private and important but its access will be protected from third parties thanks to the security of the communication protocol and between companies will sign information privacy agreements.

# ANNEXES

## 1. Code used in the Raspberry PI.

### NFC Reader Script (read_tags.py)

```
import commands
import re
from nfc import *
#import nfc
#import nfc.ndef
import socket
UDP_IP = "127.0.0.1"
UDP_PORT = 5005

def connected(tag):
    ID=str(tag)
    #print ID
    longitud=len(ID)
    posicion=ID.find("T")
    ID_code=ID[posicion+4:]
    longitud=len(ID_code)
    #result = ID_code.replace(r'?','')
    result=ID_code[:longitud-1]
    print("-------------------------------------")
    print("-------------------------------------")
    print(result)
    print("-------------------------------------")
    #print (ID_code)
    sock  =  socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
# UDP
    sock.sendto(result, (UDP_IP, UDP_PORT))
    return False

if __name__ == "__main__":
    try:
        data = read()
        data_tag=dataToAscii(data)
        connected(data_tag)
    except:
        print("-------------------------------------")
        print("Please Check the NFC TAG and try again")
        print("-------------------------------------")
```

### OPCUA Server Script (send_opc.py)

```
from opcua import Server
from random import randint
import datetime
import time
import socket
UDP_IP = "127.0.0.1"
UDP_PORT = 5005
sock = socket.socket(socket.AF_INET,socket.SOCK_DGRAM) # UDP
sock.bind((UDP_IP, UDP_PORT))

server = Server()
url = "opc.tcp://192.168.1.102:4842"
server.set_endpoint(url)
name = "OPCUA_CONNECTION"
addSpace = server.register_namespace(name)
node = server.get_objects_node()

ServerInfo = node.add_object(addSpace, "OPCUA CONNECTION")
Param = node.add_object(addSpace, "Parameters")

Id = Param.add_variable(addSpace, "ID", 0)
Loc = Param.add_variable(addSpace, "Location", 0)

Id.set_writable()
Loc.set_writable()
server.start()
print("Server Started at".format(url))

while True:
    data, addr = sock.recvfrom(1024)  # buffer size is 1024 
bytes
    received=data.decode('utf-8')
    print("---------------------------")
    print("Sent to ODOO:  ")
    print("---------------------------")
    #Ids = randint(10, 50)
    Id.set_value(data)
    location="18"
    Loc.set_value(location)

    print("code: ",received)
    print("location: ",location)
    print("---------------------------")

    #time.sleep(2)
```

## 2. Code used in the ODOO Server Virtual Machine to implement OPCUA Client and to Update the information in ODOO.

**(odoo_update_inventory.py)**

```
#Libreria para python 3
import xmlrpc.client

def odoo(code,warehouse):
    username = 'gusdelgadot@gmail.com' #the user
    password = 'admin'      #the password of the user
    db = 'EGDT'    #the database
    url="http://localhost:8069"

    common =
xmlrpc.client.ServerProxy('{}/xmlrpc/2/common'.format(url))
    ##print(common.version())
    uid = common.authenticate(db, username, password, {})
    models =
xmlrpc.client.ServerProxy('{}/xmlrpc/2/object'.format(url))

    #__________________________________
    #Read the barcode number
    barcode=code
    code_location=warehouse

    #Consult the name
    consulta_nombre=models.execute_kw(db, uid,
password,'product.template', 'search_read',[[['barcode',
'=', barcode]]],{'fields': ['name','display_name', 'type',
'qty_available','property_stock_inventory','barcode','defau
lt_code','property_stock_procurement'], 'limit': 5})
    consulta_nombre=consulta_nombre[0]

    name=consulta_nombre['name']
    print (name)
    default_code=consulta_nombre['default_code']
    print (default_code)

    product_id="["+default_code+"] "+name
    #print(product_id)
    #Consult the code of the part
    consulta_codigo=models.execute_kw(db, uid,
password,'stock.quant', 'search_read',[[['product_id', '=',
product_id]]],{'fields': ['name','display_name',
'location_id','product_id']})
    #consulta_codigo=consulta_codigo[1]
    consulta_codigo=consulta_codigo[0]
```

```
    print(consulta_codigo)
    product_id2=consulta_codigo['id']
    #print (product_id2)

    #Actualizar ID de localizacion
    models.execute_kw(db, uid, password, 'stock.quant', 
'write', [[product_id2], {'location_id': code_location}])
    #models.execute_kw(db, uid, password, 'stock.quant', 
'write', [[55], {'location_id': 19}])

    print ("Succesfully Updated")
```

**(opc_interfaz_2.py)**

```
from opcua import Client
import time
import odoo_update_inventory

try:
    #Server conection
    #url="opc.tcp://10.211.55.21:4840"  #De el servidor
    url = "opc.tcp://192.168.1.100:4842"
    client=Client(url)

    client.connect()
    print ("Client Conected")
except:
    print("Problem with the Server")

#The Loop
aux = ""
aux2 = ""
while True:

    try:
        #Node where is the ID
        Id =client.get_node("ns=2;i=3")
        Ids = Id.get_value()
        Ids=Ids.decode('utf-8')
        #print("inicio")
        #print("Ids", Ids)
        #print("aux", aux)
        #node where is the WarehouseID
        Warehous=client.get_node("ns=2;i=4")
        Warehouses=Warehous.get_value()
        Warehouses=Warehouses
        #if (Ids!=aux) and (aux2!=Warehouses):
            #Here all that have to do once
        #    print (Ids,Warehouses)
```

```
        if (Ids != aux):
            print(Ids, Warehouses)
            time.sleep(2)
            #aux=Ids
            #aux2=Warehouses
            #print ("fin")
            #print("Ids",Ids)
            #print("aux", aux)
            #odoo_update_inventory.odoo("12345","27")
            odoo_update_inventory.odoo(Ids, Warehouses)
            aux=Ids


    except:
        print("Error: ITEM NOT REGISTERED ")
        client.connect()
    #    break
```

3. **Code used in python with the Web API to check the warehouse code used to update the location.**

```
import xmlrpc.client

username = 'gusdelgadot@gmail.com' #the user
password = 'admin'      #the password of the user
db = 'EGDT'    #the database
url="http://localhost:8069"

common =
xmlrpc.client.ServerProxy('{}/xmlrpc/2/common'.format(url))
##print(common.version())
uid = common.authenticate(db, username, password, {})
models =
xmlrpc.client.ServerProxy('{}/xmlrpc/2/object'.format(url))
print("")
print("stock.pack.operation")
print(models.execute_kw(db, uid,
password,'stock.pack.operation', 'search_read',
[[['lots_visible', '=', True]]],{'fields':
['product_id','qty_done',
'location_id','location_dest_id']}))
```

**Result in the terminal of the previous code**

```
[{
    'location_dest_id': [9, 'Partner Locations/Customers'],
    'qty_done': 0.0,
    'location_id': [12, 'WH/Stock'],
    'id': 1,
    'product_id': False
```

}, {
**'location_dest_id': [9, 'Partner Locations/Customers'],**
'qty_done': 0.0,
'location_id': [27, 'Chic/Stock'],
'id': 3,
'product_id': False
}, {
**'location_dest_id': [12, 'WH/Stock'],**
'qty_done': 0.0,
'location_id': [8, 'Partner Locations/Vendors'],
'id': 4,
'product_id': [29, '[CPUi5] Processor Core i5 2.70 Ghz']
}, {
**'location_dest_id': [12, 'WH/Stock'],**
'qty_done': 0.0,
'location_id': [8, 'Partner Locations/Vendors'],
'id': 5,
'product_id': [55, '[ADPT] USB Adapter']
}, {
**'location_dest_id': [12, 'WH/Stock'],**
'qty_done': 0.0,
'location_id': [8, 'Partner Locations/Vendors'],
'id': 6,
'product_id': [16, '[M-Opt] Mouse, Optical']
}, {
**'location_dest_id': [27, 'Chic/Stock'],**
'qty_done': 0.0,
'location_id': [8, 'Partner Locations/Vendors'],
'id': 7,
'product_id': [55, '[ADPT] USB Adapter']
}, {
**'location_dest_id': [27, 'Chic/Stock'],**
'qty_done': 0.0,
'location_id': [8, 'Partner Locations/Vendors'],
'id': 8,
'product_id': [39, '[HEAD] Headset standard']
}, {
**'location_dest_id': [27, 'Chic/Stock'],**
'qty_done': 0.0,
'location_id': [8, 'Partner Locations/Vendors'],
'id': 9,
'product_id': [29, '[CPUi5] Processor Core i5 2.70 Ghz']
}, {
**'location_dest_id': [9, 'Partner Locations/Customers'],**
'qty_done': 0.0,
'location_id': [12, 'WH/Stock'],
'id': 10,

'product_id': [34, '[LAP-CUS] Laptop Customized']
}, {
**'location_dest_id': [9, 'Partner Locations/Customers'],**
'qty_done': 0.0,
'location_id': [27, 'Chic/Stock'],
'id': 11,
'product_id': [11, '[B3424] Custom Computer (kit)']
}, {
**'location_dest_id': [9, 'Partner Locations/Customers'],**
'qty_done': 0.0,
'location_id': [19, 'WH/Stock/Shelf 1'],
'id': 12,
'product_id': [12, '[A1232] iPad Mini']
}, {
**'location_dest_id': [18, 'WH/Stock/Shelf 2'],**
'qty_done': 100.0,
'location_id': [19, 'WH/Stock/Shelf 1'],
'id': 13,
'product_id': [12, '[A1232] iPad Mini']
}, {
**'location_dest_id': [19, 'WH/Stock/Shelf 1'],**
'qty_done': 1.0,
'location_id': [12, 'WH/Stock'],
'id': 14,
'product_id': [69, '[accesorios] Cargador']
}, {
**'location_dest_id': [9, 'Partner Locations/Customers'],**
'qty_done': 1.0,
'location_id': [18, 'WH/Stock/Shelf 2'],
'id': 17,
'product_id': [74, '[CH] Chair']
}, {
**'location_dest_id': [19, 'WH/Stock/Shelf 1'],**
'qty_done': 1.0,
'location_id': [18, 'WH/Stock/Shelf 2'],
'id': 18,
'product_id': [78, '[TBL] Table']
}]


## 4. ODOO WEB SERVICE API

Odoo is usually extended internally via modules, but many of its features and all of its data are also available from the outside for external analysis or integration with various tools. Part of the Model Reference API is easily available over XML and accessible from a variety of languages [22]. In this Master Thesis was used python as the programming language for the API.

**LOG IN**

Odoo requires users of the API to be authenticated before they can query most data. The xmlrpc/2/common endpoint provides meta-calls which don't require authentication, such as the authentication itself or fetching version information. To verify if the connection information is correct before trying to authenticate, the simplest call is to ask for the server's version. The authentication itself is done through the authenticate function and returns a user identifier (uid) used in authenticated calls instead of the login.

**CALLING METHODS**

*The second endpoint is xmlrpc/2/object, is used to call methods of odoo models via the execute_kw RPC function.*

Each call to execute_kw takes the following parameters:

- the database to use, a string
- the user id (retrieved through authenticate), an integer
- the user's password, a string
- the model name, a string
- the method name, a string
- an array/list of parameters passed by position
- a mapping/dict of parameters to pass by keyword (optional)

**READ RECORDS**

Record data is accessible via the read() method, which takes a list of ids (as returned by search()) and optionally a list of fields to fetch. By default, it will fetch all the fields the current user can read, which tends to be a huge amount.

# ACRONYMS

| | |
|---|---|
| API | Application Programming Interface |
| CPS | Cyber Physical Systems |
| DCOM | Distributed Component Object Model |
| EAN | European Article Number |
| ERP | Enterprise Resource Planning |
| HTTP | Hypertext Transfer Protocol |
| IDE | Integrated Development Environment |
| IOT | Internet of Things |
| IT | Information Technology |
| MES | Manufacturing Execution System. |
| NFC | Near Field Communication |
| NTAG | NFC Tag |
| OPC | OLE for Process Control |
| PLC | Programmable Logic Controller |
| RFID | Radio Frequency Identification |
| SDK | Software Development Kit |
| SSH | Secure Shell |
| TCP | Transmission Control Protocol |
| UDP | User Datagram Protocol |
| XML | Extensible Mark-up Language |

# REFERENCES

[1] "Spain: Industria Conectada 4.0," 2017.
[2] L. D. Galindo, "The Challenges of Logistics 4.0 for the Supply Chain Management and the Information Technology," 2016.
[3] S. Wang, J. Ouyang, D. Li, and C. Liu, "An Integrated Industrial Ethernet Solution for the Implementation of Smart Factory," *IEEE Access*, vol. 5, pp. 25455–25462, 2017.
[4] Ttt. Simon Brooks, Product Marketing Manager Industrial, "The Automation Pyramid: Soon to be Ancient History - TTTech." [Online]. Available: https://www.tttech.com/news-events/tech-talk/details/the-automation-pyramid/. [Accessed: 31-Jul-2018].
[5] O. F. By Stefan Hoppe, Global Vice President, "There Is No Industry 4.0 without OPC UA," *2017-06-19*. [Online]. Available: https://www.automation.com/automation-news/article/there-is-no-industry-40-without-opc-ua. [Accessed: 02-Apr-2018].
[6] ITI, "Sistemas Avanzados de eficiencia productiva para la Industria 4.0," vol. 1, p. 91, 2016.
[7] L. Domingo Galindo, "The Challenges of Logistics 4.0 for the Supply Chain Management and the Information Technology," *94*, 2016.
[8] G. Medhi, "Creating value with Industry 4.0," 2016.
[9] "Big Data Infographic Vector Illustration." [Online]. Available: https://www.shutterstock.com/es/image-vector/big-data-infographic-vector-illustration-hand-605202314. [Accessed: 31-Jul-2018].
[10] C. Gomez, "INTERNET OF THINGS AND UBIQUITOUS IP," Barcelona, 2017.
[11] "The Internet of Things."
[12] Z.-N. Wang, Y.-Y. Liu, F. Gao, M. Tang, Y. Shi, and X.-M. Zhou, "OPC UA summary," in *Proceedings of the 2nd Annual International Conference on Electronics, Electrical Engineering and Information Science (EEEIS 2016)*, 2017.
[13] W. Hardt Jun-Prof Dr-Ing Alejandro Masrur Supervisor and C. Linder, "Evaluation of OPC-UA Technology in a Car-2x Communication towards an Industry 4.0 Driven Automotive Domain Master Thesis."
[14] OPC FOUNDATIONS, "Unified Architecture - OPC Foundation." [Online]. Available: https://opcfoundation.org/about/opc-technologies/opc-ua/. [Accessed: 11-Apr-2018].
[15] "Cloud Based HMI Panels |OPC UA Platform-AIS." [Online]. Available: http://www.aispro.com/news/simplifying-internet-based-computing-in-manufacturing. [Accessed: 31-Jul-2018].
[16] Unified Automation, "UA Client-Server SDK C++ 1.0.0: OPC UA Specifications - Unified Automation GmbH." [Online]. Available: http://documentation.unified-automation.com/uasdkcpp/1.0.0/L2OpcUaSpecifications.html. [Accessed: 11-Apr-2018].
[17] Prof Salvatore Cavalieri, "OPC UA Openness, Productivity, Connectivity Unified Architecture - ppt download." [Online]. Available: http://slideplayer.com/slide/4998365/. [Accessed: 11-Apr-2018].
[18] "OPC UA Available for 120 Million Devices – OPC Connect." [Online].

Available: https://opcconnect.opcfoundation.org/2016/12/opc-ua-for-120-million-devices/. [Accessed: 31-Jul-2018].

[19] Mike Clark, “NXP begins shipping NFC tags that can wake up a host device • NFC World,” 2013. [Online]. Available: https://www.nfcworld.com/2013/08/14/325492/nxp-begins-shipping-nfc-tags-that-can-wake-up-a-host-device/. [Accessed: 03-Jul-2018].

[20] M. Kückelhaus, J. Macaulay, G. Chung, and L. Buckalew, “INTERNET OF THINGS IN LOGISTICS,” 2015.

[21] “School Inventory Management For The Busy Principal | SAFSMS Blog.” [Online]. Available: https://safsms.com/blog/managing-inventory-busy-principal/. [Accessed: 08-Jun-2018].

[22] ODOO WEB API, “Web Service API — odoo 9.0 documentation.” [Online]. Available: https://www.odoo.com/documentation/9.0/api_integration.html. [Accessed: 02-Sep-2018].